Workflows for Bringing Data Science on the Cloud/Edge Computing Continuum

Patrizio **Dazzi**[1], Valerio **Grossi**[2] and Roberto **Trasarti**[2]

[1]*Department of Computer Science, University of Pisa, Italy*

[2]*Institute of Information Science and Technologies (ISTI), National Research Council (CNR), Pisa, Italy*

Abstract

Research infrastructures play a crucial role in the development of data science. In fact, the conjunction of data, infrastructures and analytical methods enable multidisciplinary scientists and innovators to extract knowledge and to make the knowledge and experiments reusable by the scientific community, innovators providing an impact on science and society. Resources such as data and methods, help domain and data scientists to transform research in an innovation question into a responsible data-driven analytical process. On the other hand, Edge computing is a new computing paradigm that is spreading and developing at an incredible pace. Edge computing is based on the assumption that for certain applications is beneficial to bring the computation as closer as possible to data or end-users. This paper discusses about this topic by describing an approach for writing data science workflows targeting research infrastructures that encompass resources located at the edge of the network.

Keywords

Data-sharing, Data Science, Cloud Platforms, Federated Platforms

1. Introduction

The combined exploitation of data, infrastructures, and analytical methods enable multidisciplinary scientists and innovators to extract knowledge and to make the knowledge and experiments reusable by the scientific community, innovators providing an impact on science and society. Data science can support policy-making, it offers novel ways to produce high-quality and high-precision statistical information, can help to promote ethical uses of big data. Research infrastructures (RIs) play a crucial role in the advent and development of data science. Resources such as data and methods help data scientists to transform research or an innovation question into a responsible data-driven analytical process. This process is executed onto the platform, supporting experiments that yield scientific output, policy recommendations, or innovative proofs-of-concept. An infrastructure offers means to define complex *workflows*, thus bridging the gap between experts and analytical technology. As a collateral effect, experiments generate new relevant data, methods, and workflows that can be integrated into the platform by scientists, contributing to the expansion of the RI. As a drawback, the availability of data creates opportunities but also new risks. The use of data science techniques could expose sensitive traits of individual persons and invade their privacy.

SoBigData RI is a platform for the design and execution of large-scale social mining experiments, open to users with diverse backgrounds, accessible on cloud, and also exploiting super-computing facilities. All the SoBigData components are introduced for implementing data science: from raw data management to knowledge extraction, with particular attention to legal and ethical aspects. SoBigData serves a cross-disciplinary community of scientists studying all the elements of societal complexity from a data- and model-driven perspective. Moreover, pushing the FAIR (Findable, Accessible, Interoperable, Reusable) and FACT (Fair, Accountable, Confidential and Transparent) principles, furthermore SoBigData++ RI renders social mining experiments more easily designed, adjusted and repeatable by experts that are not data scientists. SoBigData++ RI moves forward from the simple awareness of ethical and legal challenges in social mining to the development of concrete tools that operationalize ethics with value-sensitive design, incorporating values and norms for privacy protection, fairness, transparency and pluralism.

From the perspective of the actual deployment on physical resources, a relevant challenge for these tools consists in providing a way to exploit those resources that can not be directly involved in the administrative domains of the research infrastructure; even if their exploitation could be beneficial for supporting the execution of the application. A notable example of such kind of resources are the Edge devices, i.e., those devices that can be exploited by using a pay-per-use approach, typical of utility computing paradigm [1], but are not part of a large centralized installation (like a Cloud) and are instead distributed on a large, dispersed, area. The aim of this approach is in fact to provide a pervasive computing infrastructure with the objective of bringing the computation as much close as possible to the data producers (e.g., sensors, cameras, etc.) and/or data consumers (e.g., users, applications, etc.). The resulting research infrastructure extended to the edge, will encompass resources of different kinds; this complex set of heterogeneous resources – having different capacities and means to access – has the potentiality to enable quite more interesting scenario at a cost of an increased complexity in its actual management. This complexity is not limited to the actual setup and operation of the platform but also impacts on the approach to adopt for developing applications that should run on top of this heterogeneous and distributed resource infrastructure.

The aim of this paper is to briefly highlight how workflow-based solutions can be properly instrumented an attempt to address the aforementioned challenges. The remaining of this paper is structured as follows. Section 2 contextualize this work by placing it in the scientific literature by presenting a few related works. Section 3 introduces the workflows as a solution for developing solutions targeting traditional research infrastructures. Section 4 presents a receipt for emending existing workflows approaches to match the peculiar needs of the aforementioned extension of the infrastructure. Finally, Section 5 draws our conclusive remarks and introduces some works that we plan to undertake in the near future.

2. Related Work

Liew et al. [2] have analyzed selected Workflow Management Systems (WMSs) that are widely used by the scientific community; among those: Airavata [3], Pegasus [4], Taverna [5], and Swift [6]. Such systems have been reviewed according to the following aspects: *(i)* processing elements,

i.e., the building blocks of workflows envisaged to be either web services or executable programs; *(ii)* coordination method, i.e., the mechanism controlling the execution of the workflow elements envisaged to be either orchestration or choreography; *(iii)* workflow representation, i.e., the specification of a workflow that can meet two goals human representation and/or computer communication; *(iv)*data processing model, i.e., the mechanism through which the processing elements process the data that can be bulk data or stream data; *(v)* optimization stage, i.e., when optimization of the workflow (if any) is expected to take place that can either be build time or run-time (e.g., data workflow processing optimization [7, 8, 9]. The aforementioned approaches are defined based on the assumption that workflows are composed of machine-executable actions, i.e., performed by agents that can be programmatically invoked. They do not address the needs, motivated by several scientific contexts, e.g., Big Data and Social Mining [10], Biodiversity and Cheminformatics domains [11, 12], of defining workflows that include "manual actions", e.g., data manipulation and adaptation using editors or shell commands. Attempts in this direction exist but embrace a fully manually-oriented approach, e.g., protocols.io [13], enabling the digital representation, publishing, and sharing of digital fully manual workflows.

The main contribution of this paper is an extension of workflow language and execution platform, whose intuition was earlier presented in [14]. HyWare was designed to enable the description of "hybrid" workflows, obtained as sequences of machine-executable and manually-executable actions. As such, the language can serve the mission of Open Science by addressing the reproducibility of digital science beyond traditional approaches in contexts where workflow actions are not entirely performed by machines. In recent years there have been several efforts in studying the most appropriate solution for structuring applications for Cloud/Edge environment. TOSCA is one of the most successful standards. As Binz [15] states, the goals of TOSCA include the automation of application deployment and the representation of the application in a cloud agnostic way. This standard has been leveraged by several products and research initiatives, e.g., BASMATI [16, 17], and Tosker [18]. Tosker works with an extended TOSCA YAML and generates a deployment plan for Docker. TOSCA has also been used with Kubernetes [19] to define application components along with their deployment and run-time adaptation on Kubernetes clusters across different countries. All these solutions are general purpose and not focus on a specific class of the application; that is instead the approach that we follow in this paper.

3. Need for ad-hoc instruments

Workflows are tools for the representation of the scientific process and the steps the researchers had to perform to execute an experiment using the e-infrastructure tools. Workflows can inherently contribute to the implementation of two data principles which are at the base of the modern data processing and analysis: *(i)* the FAIR data principles by accurately collecting, processing, and managing data and metadata on behalf of the researchers, while tracking provenance according to standards [20]; *(ii)* the FACT data principles stating that the data processing should be fair, accurate, confidential and transparent. A workflow language [21]. Moreover, workflows are digital objects in their own right, they can be published, discovered, shared, and cited for reproducibility and for scientific attribution of science like research articles,

research data, and research software. Known approaches include: *workflows as digital artefacts*: workflow files are published in a repository with bibliographic metadata (e.g. Zenodo[1], [22]) and can be possibly related to their inputs and outputs [23, 24]; *workflow-as-a-Service*: workflows are shared via a platform gateway that enables discovery and execution [25, 26, 27].

SoBigData scientists can integrate their tools for VRE-integrated reuse but cannot represent a sequence of actions in order to share it and reproduce it. We are working on equipping SoBigData VREs with a workflow allowing scientists to attach to a specific result the entire process used to obtain it. This makes the environment evolve into a living laboratory, which contains not only the methods and the results but also the experience of the researcher using the methods, and composing an analytical process with it. On the other side, our workflow has to manage the computational component needed to execute an action both not only considering federated ones [28]. In this paper, we consider only machine action, i.e., actions executable by a computational node and characterized by a description, expressed by the respective properties, but also by a standard way to invoke a third-party service and get back the results. To this aim, for each machine action class will integrate a mediator capable of invoking the external service with given action input parameters and collect the parameters to return them in accordance with the action output type. Each machine action of our workflow language (Fig. 1) includes three main aspects: *(i)* Configuration parameters: this information allows the system to instantiate a generic action class of the method to a specific instance ready to be executed. *(ii)* Execution annotation: represents a form of syntactic metadata that are directives to the workflow execution environment and the Edge computing orchestrator (i.e. memory usage, multi-core-executable, execution placement, latency constraints); for example, these annotations can be used to reduce the latency of an execution of an action under a certain value, or constraints on data transmission. [29, 30]. *(i)* Returning result: packaging the results in order for them to be available to the subsequent action instance execution.

Driving Profile Example: computing driving profiles and monitoring driving behaviors of users [31] will be done at different levels of the cloud/edge continuum: the single action class will be instantiated as: *(i)* an instance of model computation at the device level to compute the user profile using the personal data produced during its daily activities and an assessment module to check if the model holds; *(ii)* a global modeling at the cloud level which combines local models and updates it if something changes in the nodes.

4. A receipt for Workflows targeting edge computing

Workflows are effective approaches to structure applications describing scientific processes enabling the development of many data science solutions. As such, the empowerment of workflows will edge-enable a large amount of data-centric applications. To achieve this goal our approach is based on the extension of our previously proposed workflow engine [14], with an edge computing orchestrator able to properly manage the execution of workflow actions on top of edge resources.

Such an extension needs to revolve around the following aspects: *(i)* interoperability: allowing the exploitation of Edge resources by enabling the actual deployment of workflow actions at

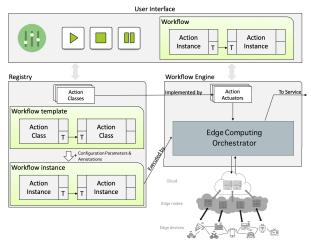---

[1]https://zenodo.org/

Figure 1: Workflow engine architecture. The classes of actions can be instantiated according to the underlying e-infrastructure and then combined into workflows.

the Edge. In spite of the many solutions proposed in the literature, an off-the-shelf solution targeting extended research infrastructures is not existing. *(ii)* resource Indexing and Discovery and representation: the workflow engine has to be provided with the ability of indexing the resources available, on which to map the workflow actions; To this end we envision the exploitation of solutions borrowed from the peer-to-peer field that we investigated in the past [32, 33] demonstrated to be quite effective solutions to this end; *(iii)* application Monitoring and Orchestration: a fundamental element for achieving an efficient exploitation of edge resources is an efficient monitoring subsystem that feeds an orchestration subsystem aimed at conducting a match-making process to provide applications with the best resources possible, among the one available; In the literature have been presented several solutions to this end, both from the domain of Clouds (e.g., Wen [34]) and Networks (e.g., Sahu [35]).

5. Conclusion

This paper discusses a solution for the extension of research infrastructure at the edge. The approach we propose is based on the identification of a the key features represented by a set of annotations that need to be integrated into a workflow engine. This work describes a first step toward a definition of a workflow language enabling the use of extended computational resources represented by Edge computing.

Acknowledgments

References

[1] C.-H. Youn, M. Chen, P. Dazzi, Cloud Broker and Cloudlet for Workflow Scheduling, Springer, 2017.

[2] C. S. Liew, M. P. Atkinson, M. Galea, T. F. Ang, P. Martin, J. I. V. Hemert, Scientific workflows: Moving across paradigms, ACM Computing Surveys 49 (2016). doi:10.1145/3012429.

[3] S. Marru, L. Gunathilake, C. Herath, P. Tangchaisin, M. Pierce, C. Mattmann, R. Singh, T. Gunarathne, E. Chinthaka, R. Gardler, A. Slominski, A. Douma, S. Perera, S. Weerawarana, Apache airavata: A framework for distributed applications and computational workflows, in: Proceedings of the 2011 ACM Workshop on Gateway Computing Environments, GCE '11, ACM, New York, NY, USA, 2011, pp. 21–28. doi:10.1145/2110486.2110490.

[4] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. F. da Silva, M. Livny, K. Wenger, Pegasus, a workflow management system for science automation, Future Generation Computer Systems 46 (2015) 17 – 35. doi:10.1016/j.future.2014.10.008.

[5] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi, C. Goble, The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud, Nucleic Acids Research 41 (2013) W557–W561. URL: http://dx.doi.org/10.1093/nar/gkt328. doi:10.1093/nar/gkt328.

[6] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, V. Nefedova, I. Raicu, T. Stef-Praun, M. Wilde, Swift: Fast, reliable, loosely coupled parallel computation, in: IEEE Congress on Services (Services 2007), 2007, pp. 199–206. doi:10.1109/SERVICES.2007.63.

[7] G. Kougka, A. Gounaris, A. Simitsis, The many faces of data-centric workflow optimization: a survey, International Journal of Data Science and Analytics 6 (2018) 81–107. doi:https://doi.org/10.1007/s41060-018-0107-0.

[8] A. Lulli, E. Carlini, P. Dazzi, C. Lucchese, L. Ricci, Fast connected components computation in large graphs by vertex pruning, IEEE Transactions on Parallel and Distributed systems 28 (2016) 760–773.

[9] M. M. Bersani, S. Distefano, L. Ferrucci, M. Mazzara, A timed semantics of workflows, in: A. Holzinger, J. Cardoso, J. Cordeiro, T. Libourel, L. A. Maciaszek, M. van Sinderen (Eds.), Software Technologies, Springer International Publishing, Cham, 2015, pp. 365–383.

[10] F. Giannotti, R. Trasarti, K. Bontcheva, V. Grossi, Sobigdata: Social mining & big data ecosystem, in: Companion of the The Web Conference 2018 on The Web Conference 2018, International World Wide Web Conferences Steering Committee, 2018, pp. 437–438.

[11] D. Garijo, P. Alper, K. Belhajjame, O. Corcho, Y. Gil, C. Goble, Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems 36 (2014) 338 – 351. doi:10.1016/j.future.2013.09.018, special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications.

[12] N. Schaduangrat, S. Lampa, S. Simeon, M. P. Gleeson, O. Spjuth, C. Nantasenamat, Towards

reproducible computational drug discovery, Journal of Cheminformatics 12 (2020) 9. URL: https://doi.org/10.1186/s13321-020-0408-x. doi:10.1186/s13321-020-0408-x.

[13] L. Teytelman, A. Stoliartchouk, L. Kindler, B. L. Hurwitz, Protocols.io: Virtual communities for protocol development and discussion, PLOS Biology 14 (2016) 1–6. URL: https://doi.org/10.1371/journal.pbio.1002538. doi:10.1371/journal.pbio.1002538.

[14] L. Candela, V. Grossi, P. Manghi, R. Trasarti, A workflow language for research e-infrastructures, International Journal of Data Science and Analytics (2021). URL: https://doi.org/10.1007/s41060-020-00237-x. doi:10.1007/s41060-020-00237-x.

[15] T. Binz, U. Breitenbücher, O. Kopp, F. Leymann, Tosca: portable automated deployment and management of cloud applications, in: Advanced Web Services, Springer, 2014, pp. 527–549.

[16] J. Altmann, B. Al-Athwari, E. Carlini, M. Coppola, P. Dazzi, A. J. Ferrer, N. Haile, Y.-W. Jung, J. Marshall, E. Pages, E. Psomakelis, G. Z. Santoso, K. Tserpes, J. Violos, Basmati: An architecture for managing cloud and edge resources for mobile users, in: C. Pham, J. Altmann, J. Á. Bañares (Eds.), Economics of Grids, Clouds, Systems, and Services, Springer International Publishing, Cham, 2017, pp. 56–66.

[17] J. Violos, V. M. de Lira, P. Dazzi, J. Altmann, B. Al-Athwari, A. Schwichtenberg, Y.-W. Jung, T. Varvarigou, K. Tserpes, User behavior and application modeling in decentralized edge cloud infrastructures, in: International Conference on the Economics of Grids, Clouds, Systems, and Services, Springer, Cham, 2017, pp. 193–203.

[18] A. Brogi, L. Rinaldi, J. Soldani, Tosker: a synergy between tosca and docker for orchestrating multicomponent applications, Software: Practice and Experience 48 (2018) 2061–2079.

[19] D. Kim, H. Muhammad, E. Kim, S. Helal, C. Lee, Tosca-based and federation-aware cloud orchestration for kubernetes container platform, Applied Sciences 9 (2019) 191.

[20] C. Goble, S. Cohen-Boulakia, S. Soiland-Reyes, D. Garijo, Y. Gil, M. R. Crusoe, K. Peters, D. Schober, Fair computational workflows, Data Intelligence 2 (2020) 108–121. URL: https://doi.org/10.1162/dint_a_00033. doi:10.1162/dint_a_00033. arXiv:https://doi.org/10.1162/dint_a_00033.

[21] B. Lepri, N. Oliver, E. Letouzé, A. Pentland, P. Vinck, Fair, transparent, and accountable algorithmic decision-making processes: The premise, the proposed solutions, and the open challenges, Philosophy & Technology 31 (2018). doi:10.1007/s13347-017-0279-x.

[22] D. D. Roure, C. Goble, R. Stevens, The design and realisation of the experimentmy virtual research environment for social sharing of workflows, Future Generation Computer Systems 25 (2009) 561 – 567. doi:10.1016/j.future.2008.06.010.

[23] D. Garijo, Y. Gil, O. Corcho, Abstract, link, publish, exploit: An end to end framework for workflow sharing, Future Generation Computer Systems 75 (2017) 271 – 283. doi:https://doi.org/10.1016/j.future.2017.01.008.

[24] A. Shaon, S. Callaghan, B. Lawrence, B. Matthews, A. Woolf, T. Osborn, C. Harpham, A linked data approach to publishing complex scientific workflows, in: 2011 IEEE Seventh International Conference on eScience, 2011, pp. 303–310. doi:10.1109/eScience.2011.49.

[25] R. Filgueira, M. Atkinson, A. Bell, I. Main, S. Boon, C. Kilburn, P. Meredith, Escience gateway stimulating collaboration in rock physics and volcanology, volume 1, 2014, pp. 187–195. doi:10.1109/eScience.2014.22.

[26] M. Danelutto, P. Dazzi, et al., A java/jini framework supporting stream parallel computations., in: PARCO, 2005, pp. 681–688.

[27] R. Baraglia, P. Dazzi, M. Mordacchini, L. Ricci, L. Alessi, Group: A gossip based building community protocol, in: Smart spaces and next generation wired/wireless networking, Springer, Berlin, Heidelberg, 2011, pp. 496–507.

[28] M. Coppola, P. Dazzi, A. Lazouski, F. Martinelli, P. Mori, J. Jensen, I. Johnson, P. Kershaw, The contrail approach to cloud federations, in: Proceedings of the International Symposium on Grids and Clouds (ISGC'12), volume 2, 2012, p. 1.

[29] M. Danelutto, P. Dazzi, Workflows on top of a macro data flow interpreter exploiting aspects, in: Making Grids Work, Springer, Boston, MA, 2008, pp. 213–224.

[30] M. Danelutto, P. Dazzi, D. Laforenza, M. Pasin, L. Presti, M. Vanneschi, Pal: High level parallel programming with java annotations, in: Proceedings of CoreGRID Integration Workshop (CIW 2006) Krakow, Poland, Academic Computer Centre CYFRONET AGH, 2006, pp. 189–200.

[31] M. Nanni, R. Trasarti, A. Monreale, V. Grossi, D. Pedreschi, Driving profiles computation and monitoring for car insurance crm, ACM Trans. Intell. Syst. Technol. 8 (2016). URL: https://doi.org/10.1145/2912148. doi:10.1145/2912148.

[32] E. Carlini, M. Coppola, P. Dazzi, D. Laforenza, S. Martinelli, L. Ricci, Service and resource discovery supports over p2p overlays, in: 2009 International Conference on Ultra Modern Telecommunications & Workshops, IEEE, 2009, pp. 1–8.

[33] R. Baraglia, P. Dazzi, B. Guidi, L. Ricci, Godel: Delaunay overlays in p2p networks via gossip, in: IEEE 12th International Conference on Peer-to-Peer Computing (P2P), IEEE, 2012, pp. 1–12.

[34] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, M. Rovatsos, Fog orchestration for internet of things services, IEEE Internet Computing 21 (2017) 16–24.

[35] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, V. Smith, On the convergence of federated optimization in heterogeneous networks, arXiv preprint arXiv:1812.06127 3 (2018).