A Strategy for Provenance Gathering in Distributed Scientific Workflows

Anderson Marinho, Cláudia Werner, Sérgio Manuel Serra da Cruz, Marta Mattoso
PESC/COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
{andymarinho, werner, serra, marta}@cos.ufrj.br

Vanessa Braganholo
DCC/IM
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
braganholo@dcc.ufrj.br

Leonardo Murta
IC
Fluminense Federal University
Niterói, Brazil
leomurta@ic.uff.br

Abstract — Running scientific workflows in distributed environments is motivating the definition of provenance approaches that are loosely coupled to the workflow system. This kind of approach is interesting because it allows both storage and access to provenance data in an integrated way, even in an environment where different workflow management systems work together. In order to provide provenance functionalities, the existing approaches overload scientists with many manually computing tasks, such as script adaptations and implementations of extra functionalities. However, when we are dealing with users who do not have such expertise (the majority of scientists do not have it), this is not a good solution. Hence, the objective of this paper is to define a provenance strategy that facilitates the gathering of provenance information in a distributed environment scenario.

Keywords - provenance; scientific workflows; distributed environment;

I. INTRODUCTION

Scientific experiments are one of the ways used by scientists to investigate phenomena with the aim of acquiring new knowledge or correct and integrate previously established knowledge [1]. In the last decades, computing has made great progresses and scientific experiments began to use computational tools to ease their execution. The experiment environment, objects, and subjects began to be simulated, rising a new category of experiments, the *in silico* experiments [2]. To support this category of experiments, the workflow concept is used to ease the experiment abstraction, allowing a structured composition of programs as a sequence of activities that aims a specific result [3]. Scientific Workflow management systems (SWfMS) ease the construction and execution of the so called "scientific workflows". A SWfMS not only makes explicit the steps of the workflow execution but also enables an automatic collection of workflow data products and their provenance information [4]. For simplicity, from now on scientific workflows will be referred to as workflows.

Provenance provides historical information about data manipulated in a workflow [5]. This historical information tells us how data products were generated, showing their transformation processes from primary input and intermediary data. The management of this kind of information is important since it provides to scientists a variety of data analysis applications. For instance, from provenance information it is possible to verify data quality of generated data products, because we can look at their ancestral data and determine if they are reliable or not. Other examples are the possibility to audit trails to verify what resources are being used, data derivation, experiment reproduction capability, responsibility attribution, among other applications [6].

To benefit from provenance information, it has to be gathered, modeled, and stored for further queries. Provenance management is an open issue that is being addressed by several SWfMS and the Provenance Challenge series. One of the open problems is related to which provenance data should be gathered and how. To address which data should be gathered, there is the initiative towards a standard model, the Open Provenance Model (OPM) [7], which defines a generic representation for provenance data. Provenance gathering becomes more complex when the workflow is executed by different execution environments. This means gathering provenance from heterogeneous sources.

According to Freire et al. [5], a provenance gathering mechanism can work at three main levels: workflow, operating system (OS), and activity. When gathering is performed at the workflow level, a SWfMS is responsible for gathering all the provenance information. Even though this is the most popular approach, this level has the disadvantage of being dependent of the SWfMS. Mechanisms that work at the operating system level use OS's functionalities for gathering provenance information (e.g.: file-system, system call tracer, etc.). The advantage of this approach is the SWfMS independency. However, the data collected by these mechanisms are fine-grained and need to be post-processed. The activity level tries to merge the best characteristics of the other two levels. At this level, each workflow activity is responsible for gathering its own provenance information. The advantage of this level is the SWfMS independency, just like the OS level, and more precise information is gathered, just like workflow level gathering. The problem of this level is the need of adaptation of preexisting activities to incorporate provenance collection functionalities.

978-0-7695-3708-5/09 $25.00 © 2009 IEEE
DOI 10.1109/SERVICES-I.2009.53

The goal of this paper is to present a provenance gathering strategy for scientific workflows that are executed in a distributed environment. This strategy was proposed to be independent of SWfMS and to demand low effort from the scientist in order to make the provenance gathering feasible.

This paper is organized as follows. Section II presents some scenarios of workflow execution in a distributed environment. Section III describes a strategy for provenance gathering according to these scenarios. Finally, Section IV concludes the paper pointing some future directions.

II. DISTRIBUTED WORKFLOW EXECUTION SCENARIOS

There are several scenarios of workflow execution in a distributed environment. Each one has its own characteristics that make provenance management difficult. According to Fig. 1, these scenarios can be classified into four types: (i) remote execution of one or more workflow activities; (ii) remote execution of a sub-workflow; (iii) remote execution of a sub-workflow by another SWfMS; and (iv) execution of two or more workflows that are part of the same experiment in distinct SWfMS.

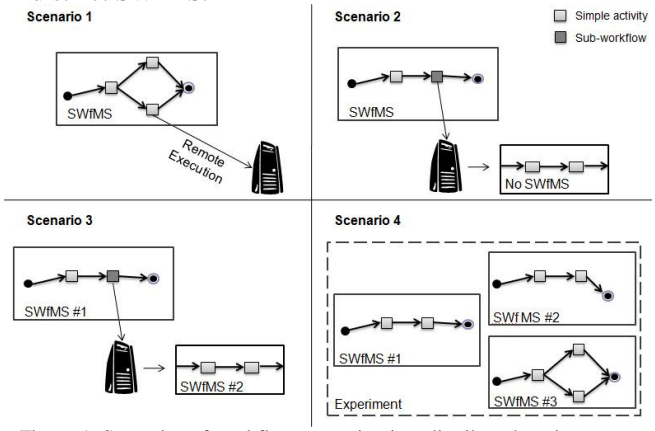

Figure 1. Scenarios of workflow execution in a distributed environment.

The first type is the simplest scenario since a SWfMS that provides a good provenance management is able for gathering provenance data even when activities are executed remotely. However, for the remaining scenarios, the use of a SWfMS is not enough to manage all the provenance information, since data can be lost. For example, in the second and third scenarios, the SWfMS that is executing the main workflow does not know about the remote execution of the sub-workflow. It does not know what activities are executed nor even what data were consumed or generated by them. The SWfMS only knows the input and output data of the whole sub-workflow. In the third scenario, the scientist has the chance of verifying this provenance information in the secondary SWfMS. Nevertheless, there is a high probability that this information is represented differently from the main SWfMS, making the analysis process more complex to the scientist. The fourth scenario is an extreme situation where the workflow of an experiment is fragmented into several smaller workflows in order to be executed in an heterogeneous environment, that has different SWfMS. In this case, each SWfMS manages provenance information in a decentralized way, meaning that each system considers provenance in a specific granularity, stores the information on a specific language, or even worse, some SWfMS do not provide a provenance solution at all. In situations like that, we can say that the experiment has a heterogeneous provenance support. This last scenario, although being an extreme situation, is becoming common in scientific experiments. This behavior is motivated by the fact that specific SWfMS have particular properties, and the adoption of different SWfMS in different regions of the workflow is more advantageous than the adoption of only one SWfMS for the workflow as a whole. For example, some workflow regions need to be executed in SWfMS that support results visualization. Other workflow regions need to be executed in SWfMS that provide grid support, and so on. An additional reason may be due to organizational issues that imply that the workflow execution should occur in several laboratories of a virtual institute.

Therefore, a strategy to solve this problem is to transfer the responsibility of provenance management to an independent system. This system would be responsible for gathering, storing, and providing integrated provenance information of an experiment. Some related work, such as [8-12] have the same concerns and share the idea of a provenance management system that is independent of a workflow system technology.

One of the main shortcomings of this strategy is that the SWfMS and a provenance management system need to communicate to exchange information. In order to make this communication feasible, some researches [9,13,14] propose a series of manual activity adaptations over the workflow definition. However, this is not a good solution from the scientist point of view. First of all, the majority of scientists have no computation skills. Besides, most workflow activities used by scientist are from third-parties, which makes their adaptation more complex. In worst cases, these activities cannot be altered. Therefore, we claim that the workflow adaptation should be done as automatic as possible, avoiding to overload scientists with computing tasks that are beyond their main goal.

III. GATHERING PROVENANCE IN A DISTRIBUTED ENVIRONMENT

As described in the previous section, gathering and managing provenance of workflows executed in distributed environments is a complex task. This is due to several SWfMS used and the provenance information generated in a heterogeneous environment. In this section, we present our strategy that has the goal of mitigating this problem. Our strategy can be seen as an enabling component for a complete provenance management approach for heterogeneous environments.

In order to conceive a provenance gathering strategy that operates in distributed environments, it is important to be independent of workflow-related technology, i.e. independent of any SWfMS, activities scheduler, execution engines, etc. This independence is guaranteed by the fact that our provenance gathering strategy focus on the activity level. At this level, as discussed in Section I, each workflow

activity is responsible for gathering its own provenance information. This provenance information is published in an integrated repository, making the analysis process easier to the scientist. However, the scientist workflow does not usually have such mechanism incorporated. Therefore, some adjustments in the workflow are necessary to allow the provenance mechanism to work. For convenience, one could delegate this task to the scientist, increasing the scientist burden for using the system. Nevertheless, this is not a good strategy. Due to that, our strategy entails the automatic workflow configuration to support the provenance gathering.

Fig. 2 presents an overview of our strategy in action. The first step that the scientist has to do is to configure the workflow to allow provenance gathering. This configuration is automatically executed, adapting each workflow activity to work with the provenance gathering mechanism. During the workflow adaptation, we also capture provenance information related to workflow development time, also known as prospective provenance [5]. After that, when the scientist runs the experiment, all provenance information related to workflow execution, also known as retrospective provenance [5], are transferred from each workflow activity to the provenance repository. Once the experiment is executed, the scientist has an integrated location to visualize and analyze the provenance data.

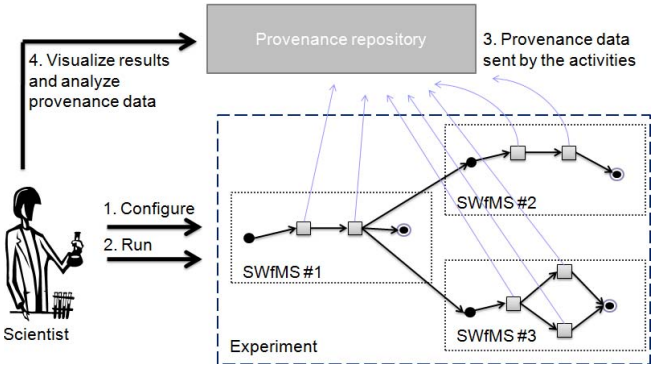

Figure 2. Overview of provenance gathering strategy in action.

By analyzing this process, some issues arise. The first one is the workflow specification, which may be written in different languages, according to the SWfMS. However, each SWfMS has its own characteristics, such as workflow language specification, execution mechanism restrictions, etc. For that reason, our strategy should be customized for the specific needs of different SWfMS, such as VisTrails [15], Kepler [16], etc. Another issue is how to adapt an activity to support the provenance mechanism. Once again, the answer depends on the SWfMS. Some SWfMS write the activity code in the workflow specification, others work with just service calls or executables. In other words, it is hard to directly adapt the workflow activity.

A solution to solve the activity adaptation problem is to make it indirectly. This can be done through the construction of wrappers that encapsulate the original activities. The wrapper adds the provenance gathering mechanism, which is responsible for gathering the information from the original activity and sending it to the provenance repository. Fig. 3 shows the structure of an adapted activity, which is composed of the provenance gathering mechanism module and the original activity. It is possible to observe that the adapted activity input and the original activity output data are transferred to the provenance gathering mechanism, which can then be transferred to the provenance repository.

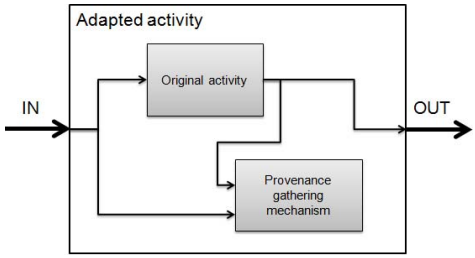

Figure 3. Wrapper structure of an adapted activity.

In order to implement the wrapper solution, we can use the composite activity concept, which is provided by most SWfMS. A composite activity aggregates activities, forming a sub-workflow. So, for each workflow activity, we create a composite activity that contains the original activity. In addition, we also include provenance gathering activities (PGA), which perform the provenance capturing. Each PGA is responsible for publishing, via services, particular provenance information, such as the beginning and ending of an activity execution, the artifacts produced, etc.

Fig. 4 illustrates an example of workflow adaptation using composite activities in Kepler. The workflow on top is the original workflow, which is composed of three activities, called "actors" in Kepler. These three activities are replaced by composite activities, which are called "composite actors" in Kepler. The adaptation of one of these activities (activity A) is illustrated in the sub-workflow at the bottom of Fig. 4. Notice that some PGA are placed before the activity A execution, and others are placed after. It depends on the type of provenance it publishes. For instance, a PGA that publishes the start of an activity execution has to be executed before the original activity in the sub-workflow. The opposite happens to a PGA that publishes the end of an activity execution. Notice also that two or more PGA can be executed in parallel to optimize the workflow execution.

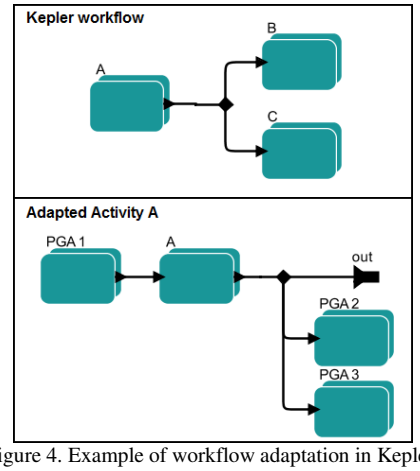

Figure 4. Example of workflow adaptation in Kepler.

IV. CONCLUSIONS

This paper presented a provenance gathering strategy that is independent of a specific SWfMS technology. This strategy was designed to enable provenance collection from different scenarios of workflow execution in distributed environments, where sometimes more than one SWfMS is used to run a single experiment. Additionally, our strategy aims at easing the scientist work. One example is the mechanism for automatically configuring the workflow to support the provenance gathering mechanism.

The adopted strategy of working with the provenance gathering mechanism at activity level can causes an overhead on workflow processing since new activities (PGA) are inserted. In addition, depending on gathered information, these PGA may also consume high network bandwidth, since they have to transfer all generated data from the workflow activities to the provenance repository. In summary, this strategy has the potential of bringing many benefits to the scientist, but also some overhead factors that have to be analyzed to guarantee the strategy applicability.

Our strategy for provenance gathering is currently being implemented as a component of a provenance management system named ProvManager. As future work, we plan to evaluate ProvManager in real scenarios, using bioinformatics and petroleum production examples.

ACKNOWLEDGMENT

The authors would like to thank CAPES and CNPq for the financial support.

REFERENCES

[1] E.B. Wilson, An Introduction to Scientific Research, Dover Publications, 1991.

[2] R. Stevens, K. Glover, C. Greenhalgh, C. Jennings, S. Pearce, P. Li, M. Radenkovic, A. Wipat, C. Tower, e J. Campus, "Performing in silico experiments on the Grid: a users perspective," Proc UK e-Science programme All Hands Conference, Nottingham, UK: 2003, pp. 2-4.

[3] D. Hollingsworth, "Workflow Management Coalition: The Workflow Reference Model," The Workflow Management Coalition, 1995.

[4] A. Goderis, U. Sattler, P. Lord, e C. Goble, "Seven Bottlenecks to Workflow Reuse and Repurposing," Lecture Notes in Computer Science, 2005, pp. 323–337.

[5] J. Freire, D. Koop, E. Santos, e C. Silva, "Provenance for Computational Tasks: A Survey," Computing in Science & Engineering, vol. 10, 2008, pp. 11-21.

[6] Y. Simmhan, B. Plale, D. Gannon. "A survey of data provenance in e-science". SIGMOD Record 2005, 34(3):31– 36.

[7] L. Moreau, J. Freire, J. Myers, J. Futrelle, e P. Paulson, *The Open Provenance Model*, Electronics and Computer Science, University of Southampton, 2007.

[8] S.M.S. da Cruz, F.S. Chirigati, R. Dahis, M.L.M. Campos, e M. Mattoso, "Using explicit control processes in distributed workflows to gather provenance," IPAW, Utah, USA: 2008, pp. 186-199.

[9] Y.L. Simmhan, B. Plale, e D. Gannon, "A Framework for Collecting Provenance in Data-Centric Scientific Workflows," International Conference on Web Services, Chicago, Illinois, USA: 2006, pp. 427-436.

[10] P. Groth, S. Jiang, S. Miles, S. Munroe, V. Tan, S. Tsasakou, e L. Moreau, An Architecture for Provenance Systems, Electronics and Computer Science, University of Southampton, 2006.

[11] P. Groth, S. Munroe, S. Miles, e L. Moreau, "Applying the Provenance Data Model to a Bioinformatics Case," High Performance Computing and Grids in Action, vol. 16, 2008, pp. 250 - 264.

[12] S.M.S. da Cruz, P.M. Barros, P.M. Bisch, M.L.M. Campos, e M. Mattoso, "Provenance Services for Distributed Workflows," Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society, 2008, pp. 526-533.

[13] S. Munroe, S. Miles, L. Moreau, e J. Vázquez-Salceda, "PrIMe: a software engineering methodology for developing provenance-aware applications," Proceedings of the 6th international workshop on Software engineering and middleware, Portland, Oregon, USA: ACM Press New York, NY, USA, 2006, pp. 39-46.

[14] C. Lin, S. Lu, Z. Lai, A. Chebotko, X. Fei, J. Hua, e F. Fotouhi, "Service-Oriented Architecture for VIEW: A Visual Scientific Workflow Management System," Services Computing 2008 (SCC '08), 2008, pp. 335-342.

[15] S.P. Callahan, J. Freire, E. Santos, C.E. Scheidegger, C.T. Silva, e H.T. Vo, "VisTrails: visualization meets data management," Proceedings of the 2006 ACM SIGMOD international conference on Management of data, Chicago, IL, USA: ACM, 2006, pp. 745-747.

[16] B. Ludascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E.A. Lee, J. Tao, e Y. Zhao, "Scientific workflow management and the Kepler system," Concurrency and Computation, vol. 18, 2006, pp. 1039-1065.