Provenance Trace Submission

Felipe Horta[1] Flavio Costa[1] Daniel de Oliveira[1] Kary Ocaña[1]
Vítor Silva[1] Jonas Dias[1] Eduardo Ogasawara[1,2] Marta Mattoso[1]
[1]COPPE/UFRJ
Rio de Janeiro, Brazil
[2]CEFET/RJ
Rio de Janeiro, Brazi

fhorta@poli.ufrj.br, { silva,flcosta,jonasdias,danielc,kary,marta}@cos.ufrj.br,
eogasawara@cefet-rj.br

SUMMARY

Data Format	XML
Data Model	Particular
Size	52 KB
Tools used for generating provenance	Chiron Parallel Workflow Engine
Application Domain	Data-Centric Scientific Workflow
Submission Group	Felipe Horta, Vítor Silva, Flavio Costa, Jonas Dias, Daniel de Oliveira, Eduardo Ogasawara, Kary Ocana, and Marta Mattoso
Contact	fhorta@poli.ufrj.br, {silva,flcosta,jonasdias,danielc,kary,marta}@cos.ufrj.br, eogasawara@cefet-rj.br
License	cc-by-nc: Creative Commons Attribution-Noncommercial 3.0 Unported License, http://creativecommons.org/licenses/by-nc/3.0/
Model Location	http://chironmodel.appspot.com/
XML Representation	http://chironmodel.appspot.com/postgres.public.xml

Categories and Subject Descriptors
H.2.8 [**Database Applications**]: Scientific databases. H.4.1 [**Office Automation**]: Workflow management. I.6.7 [**Simulation and Modeling**]: Simulation Support Systems, Environments.

General Terms
Algorithms, Languages, Experimentation Performance.

1. EXPERIENCE STATEMENT
Scientific workflows are commonly used to model and execute large-scale scientific experiments. They represent key resources for scientists and are managed by Scientific Workflow Management Systems (SWfMS). The different languages used by SWfMS may impact in the way the workflow engine executes the workflow, sometimes limiting optimization opportunities. To tackle this issue, we recently proposed a scientific workflow algebra [1]. This scientific workflow algebra is inspired by database relational algebra and it enables automatic optimization of scientific workflows to be executed in parallel in high performance computing (HPC) environments. This way, the experiments presented in this paper were executed in Chiron, a parallel scientific workflow engine implemented to support the scientific workflow algebra. Prior the execution of the workflow, Chiron stores the prospective provenance [2] of the workflow on its provenance database. In this case, each workflow is composed by several activities, and each activity consumes relations. Similarly to relational databases, a relation contains a set of fields and it is composed by a set of tuples. Each tuple in a relation contains a series of values, each one associated to a specific field. The tuples of a relation are distributed to be consumed in parallel over the computing resources according to the workflow activity. During and after the execution, the retrospective provenance [2] is stored.

In terms of provenance data model we concentrate our efforts to design a provenance data model that extends the PROV recommendation while offering the necessary formalism to represent data collected by different SWfMS and parallel workflow engines in different environments. Also, accordingly to our explanation about our workflow definition, the components of this workflow were defined as entity. In PROV representation, an entity is something digital, physical or conceptual defined in the provenance data model. The activities, in your case, are represented as an instantiation of a component. Respecting the PROV, an activity is responsible for describing an action that happens in a period of time (during workflow execution).

2. APPLICATION
Managing provenance data is a key issue for scientific experiment execution since it allows for validation and reproducibility of the results. We selected the provenance traces of two workflows related to previous works [3]. We adapted the workflows to be self-contained and reduced the input parameter space. Thus, the provenance traces contains a subset of the results of the original experiment. The first workflow is called Buzz and it process bibliographic data from the DBLP* bibliographic database. The presented provenance trace is related to a fragment of the DBLP database only. The Buzz workflow contains five activities. It maps how many times a given buzzword appears in the titles of the published articles in a year. The second workflow is called Lanczos workflow and it represents a single iteration of the Lanczos Algorithm for a random set of input matrices. The

* http://www.informatik.uni-trier.de/~ley/db/

algorithm calculates eigenvalues and eigenvectors for the input matrixes building a Krylov subspace sequence along the iterations. The provenance trace is related to a single iteration of the algorithm, thus it contains a single eigenvalue result for the algorithm. Lanczos workflow contains six activities and it starts creating the input random matrix and finishes by computing the errors associated to the calculated eigenvalues. This workflow is an adaptation of the original Lanczos workflow used in previous work and it is designed to take advantage of runtime provenance support in Chiron [4]. In Dias *et al.* [3], the iterative process of the algorithm continues until the error achieves a given value. Runtime provenance analysis allows for scientists to monitor workflow execution and to take actions before the end of the execution (*i.e.* workflow steering).

3. POSSIBLE PROVENANCE QUERIES

As soon as provenance information is available for scientists, from both Lanczos and Buzz workflows, queries can be performed at runtime to support online actions. These online actions can be related to runtime analysis of the workflow execution or to other user actions such as workflow monitoring and steering. As an example, we present a series of queries that are based on runtime provenance data.

It is possible to use queries to analyze the execution time for each activity or task in both workflows. A task is defined as an execution instantiation of activity. This type of query can be used to monitor the workflow execution and to further analyze if a specific activity execution is consuming more (or less) time than expected. If some task fails, it is possible to check for standard error output and exit status code. If a task *t* is not running, it is possible to check the status of a previous task *d* that *t* depend on. This is interesting to check data dependencies within the workflow. It is also possible to extract performance data from the collected execution times and to monitor resource utilization. Scientists can query what data was produced and check file properties like file size and directory in the storage area.

Another important set of tasks is related to the domain of the experiment. Since we can store domain data into the provenance [5], we can perform domain specific queries. For example, in the Lanczos workflow, we can assert what matrix produced a given eigenvalue or what eigenvalue was produced with the smallest error. In the Buzz workflow, it is possible to check the most frequently used term for each year. It is also possible to select the reference to the picture file containing the frequency histogram of a given word.

4. COVERAGE

PROV-O terms	Covered (Y/N)	Comments
prov:Activity	Y	n/a
prov:Entity	Y	The definition of workflow, activity, relation and field.
prov:actedOnBehalfOf	Y	A program acts on behalf of a machine
prov:startedAtTime	Y	n/a
prov:used	Y	n/a
prov:wasAssociatedWith	Y	n/a
prov:wasAttributedTo	Y	n/a
prov:wasDerivedFrom	Y	It is used to respect the dependencies between components in workflow.
prov:wasGeneratedBy	Y	n/a
prov:wasInformedBy	Y	n/a

5. CONCLUSION

A large-scale scientific experiment may involve the execution of many compute and data intensive scientific workflows. Most of these executions are executed in parallel in HPC environments such as clusters, grids and more recently clouds. We believe runtime provenance data offers valuable information to help scientists in the management of complex experiments. Thus, we provide provenance traces of two workflows designed to take advantages of runtime provenance data. Scientists can query the provenance database to monitor the workflow execution, evaluate data dependencies and analyze domain data correlating input data with produced output. We believe the provided traces are significant samples of provenance where scientists could take advantage of runtime provenance analysis.

6. ACKNOWLEDGEMENTS

The authors thank CNPq and CAPES for partially sponsoring this research, and are also grateful to the High Performance Computing Center (NACAD-COPPE/UFRJ) where the experiments were performed.

7. REFERENCES

[1] E. Ogasawara, J. Dias, D. Oliveira, F. Porto, P. Valduriez, and M. Mattoso, “An Algebraic Approach for Data-Centric Scientific Workflows,” *Proc. of VLDB Endowment*, vol. 4, no. 12, pp. 1328–1339, 2011.

[2] J. Freire, D. Koop, E. Santos, and C. T. Silva, “Provenance for Computational Tasks: A Survey,” *Computing in Science and Engineering, v.10*, no. 3, pp. 11–21, 2008.

[3] J. Dias, E. Ogasawara, D. Oliveira, F. Porto, A. Coutinho, and M. Mattoso, “Supporting Dynamic Parameter Sweep in Adaptive and User-Steered Workflow,” in *6th Workshop on Workflows in Support of Large-Scale Science*, Seattle, WA, USA, 2011, pp. 31–36.

[4] F. Costa, D. Oliveira, K. Ocana, E. Ogasawara, and M. Mattoso, “Enabling Re-Executions of Parallel Scientific Workflows Using Runtime Provenance Data. In: 4th International Provenance and Annotation Workshop,” 2012.

[5] J. Goncalves, D. Oliveira, K. A. C. S. Ocaña, E. Ogasawara, and M. Mattoso, “Using Domain-Specific Data to Enhance Scientific Workflow Steering Queries,” in *Proc. IPAW 2012*, Santa Barbara, CA, 2012.