

# **Experiences in using provenance to optimize the parallel execution of scientific workflows steered by users\***

Marta Mattoso<sup>1</sup>, Jonas Dias<sup>1</sup>, Flavio Costa<sup>1</sup>, Daniel de Oliveira<sup>2</sup>, Eduardo Ogasawara<sup>3</sup>

<sup>1</sup>COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

<sup>2</sup>Fluminense Federal University, Niteroi, Brazil

<sup>3</sup>Federal Centre for Technological Education (CEFET/RJ), Rio de Janeiro, Brazil  
{marta, jonasdias, flscosta}@cos.ufrj.br, danielcmo@ic.uff.br, eogasawara@cefet-rj.br

**Abstract.** The main advantages from using Scientific Workflow Management Systems to manage a large-scale scientific experiment are their automatic parallel execution and the improvement of result analysis through provenance data. Provenance data becomes especially useful for scientists when it is clearly associated to their domain data. In our experience, provenance data also reveals important optimizations opportunities in parallel execution and allows for user steering of workflows at run-time. The algebraic parallel execution engine is fine tuned by provenance statistics and users have explored provenance through steering support to visualize partial results from computational fluid dynamics simulations, to improve iterative uncertainty quantification applications in geophysics and to evaluate parameter setting and algorithms in several bioinformatics analyses. We discuss three of our real use cases of provenance data analysis with users from these different domains.

## **1 Introduction**

Scientific experiments based on computer simulations are usually performed by multiple combinations of programs, each of which has a set of parameters and input data. These experiments can be modeled as scientific workflows and managed by Scientific Workflow Management System (SWfMS). Several SWfMS present a workflow execution engine with parallel execution in clusters or clouds [1–4]. However, in most systems, provenance is treated as a separate component. They generate an execution log and, only when the workflow finishes execution, this log is loaded into a provenance database [5]. The provenance ends up being specific to one workflow execution and is not available for runtime user steering.

In our algebraic workflow approach [6,7], i.e. Chiron/SciCumulus, the parallel execution engine is driven by the provenance database. The algebraic workflow definition is stored in relations inside the provenance database. The engine reads them and develops an optimized execution plan, which is mapped to tasks as a MapReduce data parallel execution. However, unlike Hadoop, the workflow engine is aware of the complete dataflow, its semantics provided by the algebra and data provenance from previous executions. As the workflow is executed, this provenance database is

---

\* The research presented in this paper was partially funded by CNPq and FAPERJ grants

augmented with execution time for each task. As the same workflow (or variations) is executed, the provenance database becomes an important statistics catalog that can be queried and analyzed. Run-time optimizations can make updates to these attribute/tuples to change the execution plan and even change some parameters from the initial workflow definition. To improve provenance queries, we extract domain data and associate it to other workflow data at the provenance database using PROV-Wf [8], a specialization of W3C PROV[9].

The next section shows how users have explored our provenance steering support in real applications like computational fluid dynamics workflows [10] and phylogenetic workflows [11], to improve iterative uncertainty quantification applications in geophysics [12] and to evaluate parameter setting [13] and algorithms in several bioinformatics analyses [14].

## 2 Provenance driving workflow execution and user steering

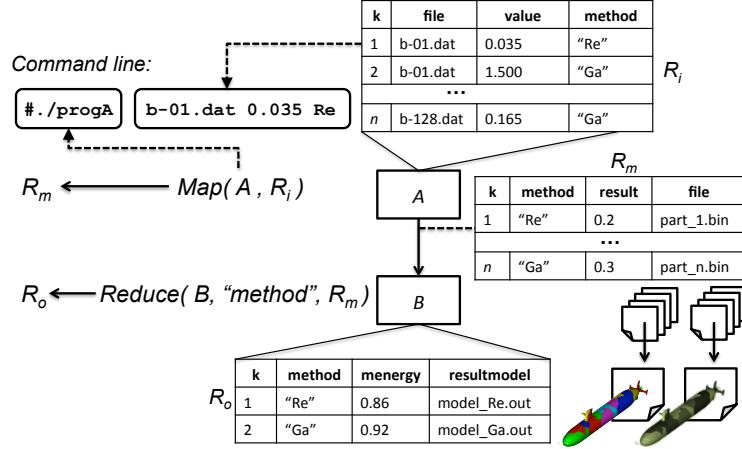
Our algebra introduces a set of operators to the relational algebra, which use both workflow activities and data as operands. An activity is ruled by an operator which has input relations, some additional operands and the result of this execution is stored at an output relation, which can be the input relation to the next workflow activity, represented as: *Output relation*  $\leftarrow$  *Operator* (*Activity*, *Additional Op*, *Input relation*). It represents algebraic expressions as in Figure 1. A scientific workflow is then a set of algebraic expressions. Using relations to represent data consumed and produced by activities allows for representing workflow data uniformly in the provenance database. Workflow parameters are now attributes of a relation and the parameter values form the tuples that are consumed and produced.

The simple workflow in Figure 1 has two activities *A* and *B*. Activity *A* is executed *n* times with different input data for three parameters: (1) a binary input file; (2) a parameter value; and (3) the method name to be used by *A*. Activity *A* generates one result value for each execution from  $R_i$  tuples. The *k* attribute relates the input with the output of *A*, which is the input of *B*. Activity *B* executes a computational simulation which can be visualized by using a third visualization activity *C*. In this case, the workflow would be represented as:

$$R_m \leftarrow Map(A, R_i); R_o \leftarrow Reduce(B, "method", R_m); R_v \leftarrow Map(C, R_o).$$

Scientists may query the relation  $R_m$  at run-time and check, if  $result > 0.5$  for any tuple. This would indicate an issue in the obtained model. Thus, they have direct access to the input and output files associated to these *result* and may update relation  $R_i$  to reset the parameter *value* and ask Chiron to process that tuple again. If scientists, based on provenance analysis, are only interested on *resultmodels* where *menergy*  $> 0.9$ , they may use relational operations in  $R_o$  to filter out unwanted results for next steps.

Provenance data also helps to improve parallel execution [15]. We have been using provenance in optimization cost functions, resource dimensioning and scheduling [16]. Chiron optimizes by identifying workflow fragments and associating parallel dataflow and dispatching strategies for each fragment [6]. Workflows can also be dynamically optimized at runtime if scientists steer the workflow based on run-time analyses of provenance data. Next, we discuss three real workflows.



**Figure 1. Algebraic workflow representation, adapted from [3].**

In [13], we present an iterative workflow that implements the Lanczos algorithm to calculate the eigenvalues of several matrixes. Lanczos quickly converges to approximate eigenvalues in both ends of the matrix spectrum. However, it is not efficient to compute the whole set of eigenvalues. Scientists usually truncate the produced series stopping the algorithm on a given iteration, but it is very difficult to know *a priori* the ideal step to stop the process. Scientists usually set the number of steps arbitrarily, what is usually prone to resource wasting. By using the steering features of Chiron/SciCumulus, scientists monitored and analyzed partial results by querying the provenance database. They now may decide when to stop the workflow. The changes scientists do in the workflow configuration immediately affect the parallel workflow execution at run-time. Our results [13] show that this approach tends to minimize the waste of resources reducing the overall execution time.

Another use case is SciPhy [11,14], a bioinformatics workflow that processes each input fasta file doing a multiple sequence alignment (MSA), an alignment conversion, a model election and building phylogenetic trees. A typical parallel execution of SciPhy may last for days, especially because scientists usually explore several MSA programs. Such programs are interchangeable but the quality of results depends on some characteristics of input data. When scientists can query the provenance database and partial results, they can evaluate that a given MSA program may not be adequate for their input dataset. For example, after 12 hours of execution, the scientist receives a notification saying that the execution may last more than expected and this is probably because the MSA activity is running slower than expected. In most current approaches, scientists cannot change the MSA program unless they abort the workflow and start it again with the new program. Using the steering features of Chiron/SciCumulus, scientists can interfere in the workflow execution to change the MSA activity (for example, use ProbCons program instead of MAFFT) [14].

A third use case is a workflow for uncertainty quantification (UQ) in geophysics [12]. UQ measures the reliability of numerical simulations, exploring the input model using stochastically varied input to, at the end, ensemble an average result and its standard deviation. The size of the exploration is associated with the precision that

scientists desire. However it is also difficult to predict the ideal precision prior the execution, because it depends on the characteristics of input data. By querying provenance data, scientists evaluate the convergence of the method by grouping and aggregating data from thousands of explorations within the same workflow. If satisfactory they may ask for the conclusion of the execution or they may even change the precision criteria, or filter values to keep the workflow running.

In addition to these three classes of applications, similar improvements have been obtained in text mining [7], oil & gas, and a large set of bioinformatics workflows through provenance runtime queries. We are now working on suggesting workflow data and algorithm configuration using provenance from several previous executions.

## References

- [1] J.M. Wozniak, T.G. Armstrong, M. Wilde, D.S. Katz, E. Lusk, e I.T. Foster, 2013, Swift/T: Large-Scale Application Composition via Distributed-Memory Dataflow Processing, In: *CCGrid'13*, p. 95–102
- [2] E. Deelman, G. Mehta, G. Singh, M.-H. Su, e K. Vahi, 2007, "Pegasus: Mapping Large-Scale Workflows to Distributed Resources", *Workflows for e-Science*, Springer, p. 376–394.
- [3] E. Ogasawara, J. Dias, V. Silva, F. Chirigati, D. Oliveira, F. Porto, P. Valduriez, e M. Mattoso, 2013, Chiron: A Parallel Engine for Algebraic Scientific Workflows, *Concurrency and Computation*, v. 25, n. 16, p. 2327–2341.
- [4] D. Oliveira, E. Ogasawara, F. Baião, e M. Mattoso, 2010, SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows, In: *IEEE 3rd International Conference on Cloud Computing*, p. 378–385
- [5] J. Freire, D. Koop, E. Santos, e C.T. Silva, 2008, Provenance for Computational Tasks: A Survey, *Computing in Science and Engineering*, v.10, n. 3, p. 11–21.
- [6] E. Ogasawara, J. Dias, D. Oliveira, F. Porto, P. Valduriez, e M. Mattoso, 2011, An Algebraic Approach for Data-Centric Scientific Workflows, *Proc. of VLDB Endowment*, v. 4, n. 12, p. 1328–1339.
- [7] J. Dias, E. Ogasawara, D. de Oliveira, F. Porto, P. Valduriez, e M. Mattoso, 2013, Algebraic dataflows for big data analysis, In: *2013 IEEE International Conference on Big Data*, p. 150–155
- [8] F. Costa, V. Silva, D. de Oliveira, K. Ocaña, E. Ogasawara, J. Dias, e M. Mattoso, 2013, Capturing and Querying Workflow Runtime Provenance with PROV: A Practical Approach, In: *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, p. 282–289
- [9] P. Groth, L. Moreau, 2013, W3C PROV- An Overview of the PROV Family of Documents, In: <http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/> (last accessed in May 2014)
- [10] F. Horta, J. Dias, R. Elias, D. Oliveira, A.L.G.A. Coutinho, e M. Mattoso, 2013, Prov-Vis: Large-Scale Scientific Data Visualization Using Provenance (Poster), In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC'13*
- [11] K.A.C.S. Ocaña, D. Oliveira, J. Dias, E. Ogasawara, e M. Mattoso, 2011, Optimizing Phylogenetic Analysis Using SciHMM Cloud-based Scientific Workflow, In: *2011 IEEE Seventh International Conference on e-Science (e-Science)*, p. 190–197
- [12] G. Guerra, F.A. Rochinha, R. Elias, D. de Oliveira, E. Ogasawara, J.F. Dias, M. Mattoso, e A.L.G.A. Coutinho, 2012, Uncertainty Quantification in Computational Predictive Models for Fluid Dynamics Using Workflow Management Engine, *Int. Journal for Uncertainty Quantification*, v. 2, n. 1, p. 53–71.
- [13] J. Dias, E. Ogasawara, D. Oliveira, F. Porto, A. Coutinho, e M. Mattoso, 2011, Supporting Dynamic Parameter Sweep in Adaptive and User-Steered Workflow, In: *6th WORKS, SC'11*, p. 31–36
- [14] I. Santos, J. Dias, D. Oliveira, E. Ogasawara, K. Ocaña, e M. Mattoso, 2013, Runtime Dynamic Structural Changes of Scientific Workflows in Clouds, In: *Proceedings of the IEEE/ACM International Workshop on Clouds and (eScience) Applications Management - CloudAM, UCC'13*, p. 417-422
- [15] L.M.R. Gadelha,Jr, M. Wilde, M. Mattoso, e I. Foster, 2011, Exploring Provenance in High Performance Scientific Computing, In: *Proc High Performance Computing Meets Databases, SC'11*, p. 17–20
- [16] D. Oliveira, K.A.C.S. Ocaña, E. Ogasawara, J. Dias, J. Gonçalves, F. Baião, and M. Mattoso, 2013, Performance evaluation of parallel strategies in public clouds: A study with phylogenomic workflows, *Future Generation Computer Systems*, v. 29, n. 7, p. 1816–1825.