Capturing Distributed Provenance Metadata from Cloud-Based Scientific Workflows

Sérgio Manuel Serra da Cruz, Carlos Eduardo Paulino, Daniel de Oliveira,
Maria Luiza Machado Campos and Marta Mattoso

Universidade Federal do Rio de Janeiro, Brazil
{serra,kdu danielc,marta}@cos.ufrj.br, mluiza@nce.ufrj.br

Abstract. Workflows are scientific abstractions used in the modeling of scientific experiments. High performance computing environments such as clusters and grids are often required to run the experiments. Cloud computing is starting to be adopted by the scientific community. However, the cloud environment is still incipient in collecting and recording retrospective workflow provenance. This article presents an approach to capturing distributed provenance metadata from cloud-based scientific workflows. The approach was implemented through an evolution of the Matrioshka architecture that was refactored for cloud environments. Preliminary results show that provenance metadata captured from the virtual components running at the cloud can aid scientists to manage and reproduce their large scale in silico experiments.

Categories and Subject Descriptors: H. Information Systems [**H.3. Information storage and retrieval**]: Databases

Keywords: Provenance, Scientific Workflows, Cloud Computing, Metadata

1. INTRODUCTION

Over the last years the e-Science field has evolved in a fast pace. Most of existing e-Science experiments deal with large volumes of data [Hey and Tansley 2009]. These experiments are also called in silico experiments [Taylor et al. 2007]. Frequently these experiments need to be executed in High Performance Computing (HPC) environments, such as clusters, grids [Kesselman and Foster 1998] and more recently, clouds [Vaquero et al. 2009]. Clouds are already being adopted as a new computational environment for scientific applications [Hoffa et al. 2008]. Clouds present several advantages for e-Science, specially the elasticity and the availability of resources. In other words, if scientists need more resources (machines or storage, for example), they just have to request that resources to the cloud provider and they will be available. Due to those characteristics, many scientists are already moving their experiments from local and private environments to the cloud [Hey and Tansley 2009; Hoffa et al. 2008; Matsunaga et al. 2008].

In silico experiments are represented by a chain of activities where each activity is mapped to an executable code (a program or a script), creating a coherent flow of data and controls, where the output of a specific activity is the input of the next activity in the flow. This flow of activities is named Scientific Workflow. Over the last years, scientific workflows became a de facto standard for modeling in silico scientific experiments [Mattoso et al. 2010]. Scientific workflows declaratively capture the activities of a scientific experiment and the dependencies between them. Such activities are represented as components (*e.g.*, command line programs) that define the computations that should take place. This data flow can be managed in *an ad-hoc* way, but it is more adequately handled by complex engines called Scientific Workflow Management Systems (SWfMS) [Taylor et al. 2007],

which offer a computational support for a scientist to define, execute, and monitor scientific workflows either locally or in remote environments. There are many types of SWfMS such as Kepler [Altintas et al. 2004], and VisTrails [Callahan et al. 2006], each one with powerful graphical interfaces and mechanisms to represent and execute scientific workflows. However, none of these systems offer cloud support, even those that are focused on HPC support, such as Pegasus.

The focus of this article is on the execution phase of scientific workflow life cycle [Mattoso et al. 2010]. To be considered as "scientific" by the community, a scientific experiment has to be able to be reproduced under the same conditions, even in different environments. In fact, descriptors associated to the workflow such as its definition, consumed and produced data during the execution are fundamental issues to consider the experiment valid, consistent and reproducible by a third party [Freire et al. 2008; Cruz et al. 2008]. This category of descriptors is called provenance metadata. Provenance (also referred as lineage or pedigree) represents the ancestry of an object [Freire et al. 2008]. Provenance of an object, such as a data product, contains information about the process used to derive the object, in this case the data related to the scientific workflow. It provides important documentation that is essential to preserve the data, to determine their quality and authorship, and to reproduce as well as to interpret and validate the associated scientific results generated by large scale scientific experiments.

We can find in literature some approaches that aim at capturing and managing provenance metadata in distributed environments. However, most of these approaches are focused on clusters and grids. One example is Matrioshka [Cruz et al. 2008]. Matrioshka aims at capturing and providing provenance metadata of scientific experiments executed in those environments. Although Matrioshka is a step forward to collect provenance metadata from distributed environments, it was initially designed for clusters and grids, not clouds. Clouds present specific characteristics such as virtualization of resources, access methods, and so on.

This article proposes an approach for the problem of capturing distributed provenance metadata in cloud environments. It describes the adaptation and effective use of the Martrioshka architecture when capturing provenance metadata in workflows executed in distributed cloud environments. In addition, it also presents a model for storing specific cloud provenance metadata. This article also presents a case study in the domain of Text Mining (TM) modeled and executed in SWfMS VisTrails. The provenance model was extended to comprise cloud specific metadata and to follow the Open Provenance Model (OPM) recommendation [Moreau et al. 2008] which proposes an agnostic representation of provenance. The chosen environment for running the experiments present in this article was the IBM cloud[1] and all components were developed using Java 1.5. and IBM DB2 9.7 for database support.

This article is organized in four sections besides this introduction. Section 2 briefly describes the concepts of cloud computing and provenance. It also discusses related work. Section 3 presents the Matrioshka architecture extended for cloud environments. Section 4 introduces and analyzes the case study of a scientific workflow for TM process. Finally, Section 5 concludes the article and points to some future work.

2. CLOUD COMPUTING AND PROVENANCE

Cloud computing has emerged as a platform for large scale data intensive computation. Its ability to provide a flexible and on-demand computing infrastructure with large scalability enables the distribution of the processing among a large number of computing nodes. According to [Foster et al. 2008] detailed the key differences between grids and clouds, defining the cloud computing as "an infrastructure of computing, provided on demand, which provides communication and control, being served by a network, in a shared and dynamically scalable way" [Hoefer and Karagiannis 2010] classify

[1]IBM http://www.ibm.com/cloud-computing/us/en/

and describe the key characteristics of cloud environments according to an e-Science perspective. One advantage of using clouds on scientific experiments is to provide scientists with access to a wide variety of resources without having to acquire and configure the computing infrastructure. Examples of in silico experiments adapted to clouds are: the Sloan Digital Sky Survey project and Berkeley Water Center [Hey and Tansley 2009]. Another feature common to these projects is the use of scientific workflows and SWfMS.

Consequently, with the execution of workflows in clouds arises the need to collect provenance metadata in cloud environments, it is necessary to ensure the reproducibility of these experiments. Without this metadata, the experiment, its evaluation and reproduction is compromised. For example, generally an execution in cloud environments occurs transparently to the scientist, i.e., the cloud infrastructure behaves like a "black box". Therefore it is critical to scientists to know what the parameters that have been used and what data products were generated in each execution of a given workflow. The capture and management of provenance metadata in distributed environments still pose an open question [Freire et al. 2008; Mattoso et al. 2010]. For example, in a cloud, the more data needs to be transmitted through the Internet the more susceptible to failure they are. This way, cloud environments, similarly to grids and clusters, need to capture and store provenance metadata. For this reason, the approach described in this article stores the provenance metadata in the cloud itself and they can be recovered afterwards.

To the best of the authors' knowledge, none of the existing cloud environments offer native support to collect provenance and any other means to store provenance metadata produced by in silico experiments. However, there are some works that highlight the importance of the subject. For instance, [Muniswamy-Reddy et al. 2009], where the authors discuss some alternatives to storage of provenance using cloud computing services offered by Amazon EC2[2] and using the PASS system. The PASS system is also a system that collects and stores provenance from distributed environments, but they are intensely involved in the collection of provenance on the generated files, unlike Matrioshka, which collects provenance metadata about processes and the execution environment, archiving them in a provenance repository. PASS proposes using three specific architectures using storage structures native to the Amazon EC2, Simple Storage Service (S3), SimpleDB and Simple Queuing Service (SQS).

3. CAPTURING DISTRIBUTED PROVENANCE METADATA ON THE CLOUD

This section describes the proposed approach to capture and store provenance metadata generated by scientific workflows running on cloud environments. The original proposal of Matrioshka was focused on overcoming some limitations of existing SWfMS in relation to metadata collection from distributed sources. Matrioshka acts as an additional layer operating regardless of the SWfMS used to enact a scientific workflow. This novel approach aims to minimize the possibility of silos of isolated provenance metadata, that is, it enables scientists to bind the provenance metadata collected from the distributed execution environment in a single database schema.

3.1 Matrioshka Architecture

The architecture developed by Cruz et al. [Cruz et al. 2008], was conceived to operate on clusters and grid environments and operated de-coupled from SWfMS, therefore, using native services from those environments, for example, process schedulers, queue managers, among others. Thus, it does not support features provided by a cloud computing environment, such as elasticity of resources, virtualization, and independence of location, among others. Matrioshka was now refactored to operate independently of the infrastructure provided by cloud providers. The architecture is composed of several components: Provenance Broker, Provenance Eavesdrop and Provenance Repository. To operate

[2]AMAZON EC2 http://aws.amazon.com/ec2/

satisfactorily in the cloud it is necessary not only to change its former database schema, but also to extend its functionality by adding two new components: Dispatcher that operates locally within the SWfMS orchestrating the workflow and the Execution Broker that operates on the cloud. The architecture is depicted in Figure 1. Each one of these components is detailed as follows.

The Provenance Broker component is responsible for receiving the provenance metadata descriptors captured by provenance gathering mechanisms, *i.e.*, the descriptors related to the execution of workflow activities on the cloud environment, The Provenance Broker stores the provenance on the metadata repository. When the execution of some activity of the workflow occurs in the cloud, the components Provenance Broker and Provenance Eavesdrop are invoked by the Dispatcher component that operates at the SWfMS layer. The Provenance Broker receives the data submitted by the Dispatcher and them the metadata captured by the Provenance Eavesdrop on the cloud . Upon receiving these metadata, the Provenance Broker persists in a data schema stored in the cloud. The Provenance Eavesdrop component performs the task of collecting the metadata generated by the activities and also the ones produced at the remote execution environment. The Broker and Eavesdrop are remote components and that works with heterogeneous metadata produced by the cloud virtual instances, such metadata can be generated from various sources, *i.e.*, running processes, files or operational information used or produced by the running virtual instances. The metadata repository stores not only the data associated with the executions of the scientific workflow, but also the metadata collected by the components. The provenance data schema is depicted in Figure 2.

The Dispatcher is a component that is executed at the local layer and should be included in the definition of the scientific workflow as an ordinary task. The Dispatcher sends remote calls of a given local activity to the Execution Broker, which invokes the activity at a given virtual instance on the cloud. The scientific program invoked by the Execution Broker must have been already installed at the virtual instances in which the scientist must have right access. Using the Dispatcher component, the scientist may set the parameters for accessing and using the virtual instances, such as your login/password, number of instances to be used, name of programs to be executed, input data and input parameters of the remote programs, among others. These parameters are stored in a manifest file in XML. The manifest contains specifications to access settings to instances of the cloud, it also may host information about the experiment itself. The manifest has the advantage of being technologically agnostic in terms of operating systems. Moreover, it also represents a set of metadata associated with retrospective provenance [Freire et al. 2008] execution of an activity of a workflow in the cloud.

The Execution Broker is a component that triggers the execution of remote activity at cloud virtual instances and when the execution is completed, returns the control to the Dispatcher component so, the workflow may continue the execution of other local activities. Figure 1 presents a conceptual representation of the Matrioshka architecture refactored for the cloud environments. The exchange of messages between the local environment and the cloud instances are performed using a secure tunneling, that uses the SSH protocol, allowing data and metadata transfer from inbound and outbound remote activities.

3.2 Provenance Schema

The provenance data schema is based on [Cruz et al. 2010] but encompasses cloud specific features, such as the concepts of virtualization and elasticity of resources, billing and service usage metering (*i.e.*, pay per use). This schema allows a scientist to query data about the workflow, the cloud provider and users involved with the execution of an *in silico* experiment. Moreover, by using instances with different configurations, a scientist must know in which instance data products of workflow activities were generated, what were the processing conditions, which resources were consumed, software versions used, among others. The new schema takes into account the latest recommendation from OPM (version 1.1). OPM aims at facilitating the interoperability of metadata that comes from heterogeneous environments and expresses the causal relationships between Processes, Agents and Artifacts in the

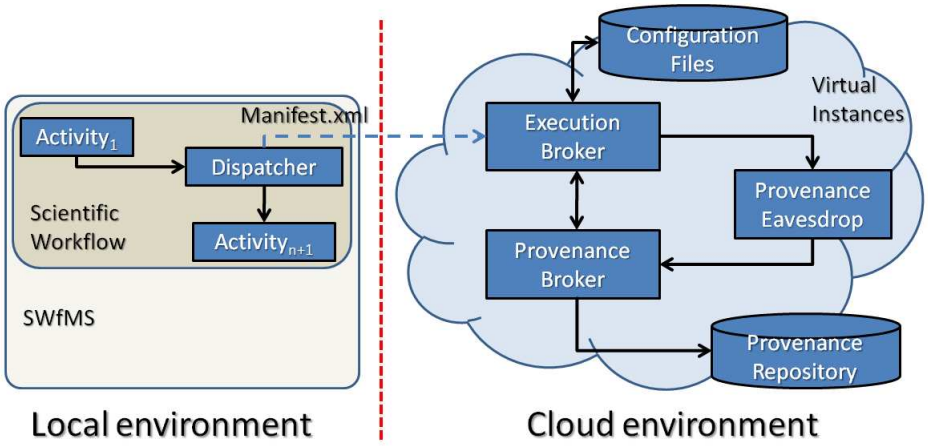

Fig. 1. The Matriohska Architecture adapted to cloud environments

existing workflows. OPM is a reference model that is not directly instanced into a database schema.

Several changes were performed on the original data schema because we had some descriptors related to cluster and grid environments, such as cluster nodes, processors and details of jobs. Besides, there was no correlation with the OPM specification. However, when we moved the schema to the cloud environment, many of these metadata lost their meaning. For example, the concept of node is replaced by virtual machines (virtual instances) made available by providers and each instance can be different from another with regard to hardware and software configurations.

The provenance data schema is represented as a UML class diagram in Figure 2, and it is the result of an initial survey [Cruz et al. 2009] about which provenance metadata have to be captured by different provenance gathering mechanisms. The data model consists of four main parts (colored to ease the understanding): (i) elements that represent the processes which are distributed in the instances of the cloud, for example, workflow activities (light blue), (ii) elements that represent the scientists responsible to workflow execution (light red); (iii) elements that represent the artifacts and the computational resources used in a given workflow execution, and finally (light orange), (iv) elements representing information related to the temporality of the workflow and its activities execution (light green).

The schema followed the recommendation of the OPM, the classes CloudOutput and CloudInstance correspond to conceptual representation of an OPM-artifact, having the same semantics, for example, i.e., both represent structures in digital computing systems (parameters, databases, files, instances, images, etc.). The class CloudActivity is mapped as an OPM-Process. A process represents one or more actions that operate on artifacts or produce new artifacts. The classes CloudUser and CloudProvider represent an OPM-Agent. An agent is the element that catalyzes, enable, control or affect the execution of a given process. The classes CloudUserWorkflow, CloudUserInstance and CloudActivityInstance are OPM-Roles. A role determines and correlates the function of an agent or an artifact in a given process. Finally, the CloudExecution class represents the moment of execution of a process on the cloud.

4. CASE STUDY: TEXT MINING SCIENTIFIC WORKFLOW

Text Mining (TM) is a process that aims to find hidden knowledge from texts and present it in a concise way. Thus, we can view TM as a key component for e-Science, and it is composed by three major phases that are named pre-processing, mining and post-processing.

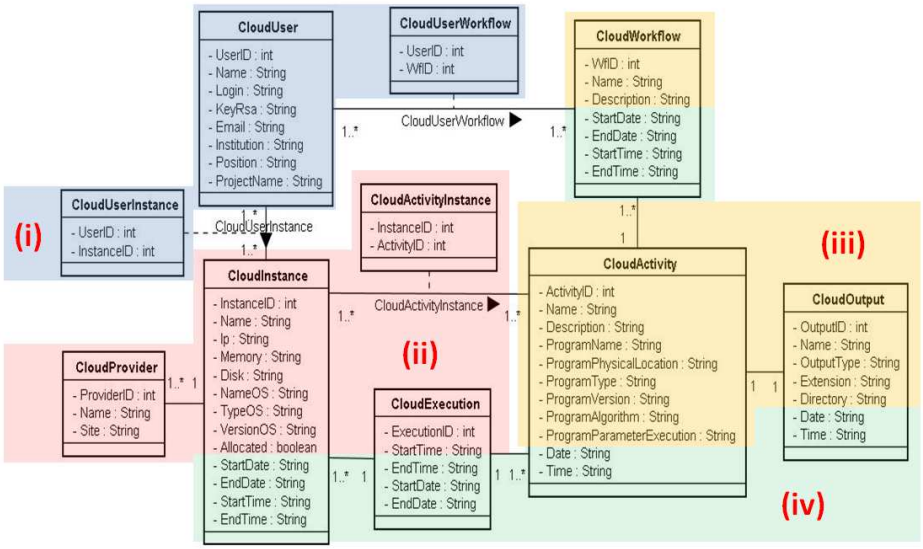

Fig. 2. Matriohska provenance data schema (colors and numbers represents the different parts of the schema) adapted from [Cruz et al. 2008]

The pre-processing phase is responsible to prepare the raw text for mining. Once the objective for the TM process is known, the text collection must be assembled. The input collection normally has noisy and unnecessary data. The source text collection must be then cleaned and prepared. Stop words removal [Dragut et al. 2009] and stemming [Korenius et al. 2004] are classical examples of pre-processing tasks.

The mining phase is the main phase of the entire TM process. It is the responsible to deriving patterns and models from pre-processed data. The mining phase is divided into many tasks also named mining tasks. Typical text mining tasks include, among others, text categorization, text clustering, concept/entity extraction, and document summarization[Fan et al. 2006].

The post-processing phase is responsible for preparing the patterns and models generated by the mining step for evaluation and visualization. The post-processing phase is divided in many functions also called post-processing functions. Each one of these functions performs a different role on the post-processing phase. TM poses as an interactive and iterative process. The TM phases may be adapted and re-executed as many times as needed. For example, to evaluate a model generated by the mining phase may be necessary to re-execute the entire mining phase in order to tune some parameters of a determined algorithm. This way, a new result is generated and the scientist may compare the two approaches.

Since this is a first viability study using a cloud environment, the workflow of TM was not entirely executed. The main focus of this study was to execute the pre-processing phase of the workflow that prepares data to be mined. The TM scientific workflow was modeled by [Oliveira et al. 2008] using VisTrails SWfMS and executed on the IBM cloud as presented in Figure 3. The collected provenance data is stored by a DB2 instance also hosted on the IBM cloud. The executed workflow has three main activities: data cleaning (stop word removal), word counting and generation of frequency table (which contains the relation words x document). When the execution finishes, the output is a CSV file that contains all frequencies off all processed collections. This file is generated on the virtual machines and transferred a posteriori to the scientists' desktop.

By executing this case study we could test the distributed execution of TM workflows on the cloud and capture important provenance metadata associated to these executions. This metadata

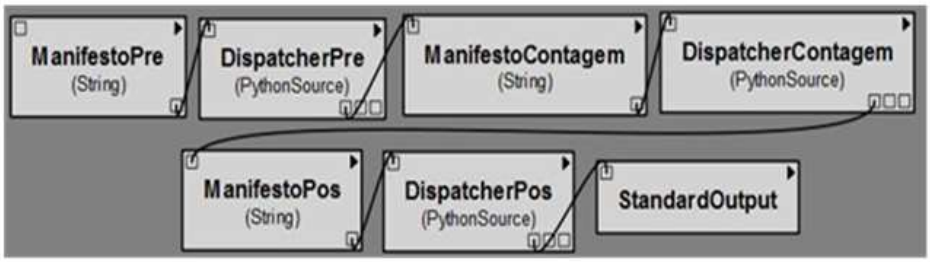

Fig. 3. Text Mining Workflow modeled in the SWfMS VisTrails adapted from [Oliveira et al. 2007]

Data Products	IP-v4 address	Directory	IP-v4 Database
1277437896654.saida.csv	129.33.196.203	/bigua/1277437896654/output	129.33.197.8 (DB2)
1277438573096.saida.csv	129.33.196.196	/bigua/1277438573096/output	129.33.197.8 (DB2)
1277438609051.saida.csv	129.33.197.23	/bigua/1277438609051/output	129.33.197.8 (DB2)
1277437897142.saida.csv	129.33.196.76	/bigua/1277437897142/output	129.33.197.8 (DB2)
1277438608846.saida.csv	129.33.195.84	/bigua/1277438608846/output	129.33.197.8 (DB2)

Fig. 4. Provenance metadata captured in the TM workflow execution on IBM cloud

includes IDs of data products consumed and produced in the course of the workflow executions, and ids of the virtual machines used to execute these workflows (IP-v4 address). In addition, it was possible to identify the instance and the type of database management system that stores the provenance metadata. All users were also identified and associated to the workflow execution in the provenance schema. In this first case study the TM workflow processed 100 documents in PDF format. This collection was distributed among five instances (virtual machines) on the IBM cloud. In each instance were uploaded 20 PDF documents. Figure 4 presents an excerpt of the captured retrospective provenance metadata by Matrioshka components. It is important to highlight that this type of provenance metadata cannot be captured with the aid of existing SWfMS.

IP-v4 addresses of the cloud instances are captured in the CloudInstance entity. All information related to produced and consumed data products in the user directory are found in the CloudOutput entity. Due to space restrictions in this article, the rest of the captured metadata was suppressed. Only the most significant and representative metadata were presented. Based on these metadata it is possible for the scientist to discover, for example, in which Virtual machines are stored the generated data products related to a specific execution of a scientific workflow.

5. CONCLUSIONS

Cloud computing presents an innovative alternative for running experiments based on scientific workflows that require distributed computing environments, primarily because of elasticity and high availability of resources. However, at the moment SWfMS provide no specific support to the execution of workflows on the clouds, especially when dealing with distribution of the activities and with the collection and storage of provenance metadata. This article describes how the Matriohska architecture was used to collect provenance metadata of scientific workflows executed in cloud environments. Moreover, we present a novel data schema that follows the OPM recommendation.

Despite being a work in progress, our initial results are promising. We are able to capture an initial set of metadata that could not be collected by existing provenance mechanisms offered by local SWfMS. As future work, we will evaluate the scalability of our solution and also will verify the performance overhead of the architecture. Moreover, further studies will be promoted to allow the integration between the data schema presented on Figure 2 with the provenance data schema offered by (local) SWfMS.

REFERENCES

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., and Mock, S. Kepler: an extensible system for design and execution of scientific workflows. In *Proceedings of International Conference on Scientific and Statistical Database Management*. Santorini Island, Greece, pp. 423 – 424, 2004.

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. VisTrails: visualization meets data management. In *Proceedings of ACM SIGMOD International Conference on Management of Data*. Chicago, USA, pp. 745–747, 2006.

Cruz, S. M. S., Barros, P. M., Bisch, P. M., Campos, M. L. M., and Mattoso, M. A provenance-based approach to resource discovery in distributed molecular dynamics workflows. In *Proceedings of International Conference on Resource Discovery*. Paris, France, pp. 66–80, 2010.

Cruz, S. M. S. d., Barros, P. M., Bisch, P. M., Campos, M. L. M., and Mattoso, M. Provenance services for distributed workflows. In *Proceedings of IEEE International Symposium on Cluster Computing and the Grid*. Washington, DC, USA, pp. 526–533, 2008.

Cruz, S. M. S. d., Campos, M. L. M., and Mattoso, M. Towards a taxonomy of provenance in scientific workflow management systems. In *Proceedings of Congress on Services*. Washington, DC, USA, pp. 259–266, 2009.

Dragut, E., Fang, F., Sistla, P., Yu, C., and Meng, W. Stop word and related problems in web interface integration. *Proc. VLDB Endow.* 2 (1): 349–360, 2009.

Fan, W., Wallace, L., Rich, S., and Zhang, Z. Tapping the power of text mining. *Commun. ACM* 49 (9): 76–82, September, 2006.

Foster, I., Zhao, Y., Raicu, I., and Lu, S. Cloud computing and grid computing 360-degree compared. In *Proceedings of Grid Computing Environments Workshop*. Austin, TX , USA, pp. 1 –10, 2008.

Freire, J., Koop, D., Santos, E., and Silva, C. T. Provenance for computational tasks: A survey. *Computing in Science and Engineering* 10 (3): 11–21, May, 2008.

Hey, T. and Tansley, S., editors. *The Fouth Paradigm:Data-Intensive Scientific Discovery*. Microsfot, 2009.

Hoefer, C. and Karagiannis, G. Taxonomy of cloud computing services. In *Proceedings of IEEE GLOBECOM Workshops*. Miami, FL, USA, pp. 1345 –1350, 2010.

Hoffa, C., Mehta, G., Freeman, T., Deelman, E., Keahey, K., Berriman, B., and Good, J. On the use of cloud computing for scientific workflows. In *Proceedings of IEEE International Conference on eScience*. Indiana, IN, USA, pp. 640 –645, 2008.

Kesselman, C. and Foster, I. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 1998.

Korenius, T., Laurikkala, J., Järvelin, K., and Juhola, M. Stemming and lemmatization in the clustering of finnish text documents. In *Proceedings of ACM International Conference on Information and Knowledge Management*. Washington, DC, USA, pp. 625–633, 2004.

Matsunaga, A., Tsugawa, M., and Fortes, J. CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications. In *Proceedings of Fourth IEEE International Conference on eScience*. Indianapolis, USA, pp. 222–229, 2008.

Mattoso, M., Werner, C., Travassos, G. H., Braganholo, V., Ogasawara, E., Oliveira, D. D., Cruz, S. M., Martinho, W., and Murta, L. Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management* 5 (1): 79–92, 2010.

Moreau, L., Freire, J., Futrelle, J., Mcgrath, R. E., Myers, J., and Paulson, P. Provenance and annotation of data and processes. Springer-Verlag, Berlin, Heidelberg, The Open Provenance Model: An Overview, pp. 323–326, 2008.

Muniswamy-Reddy, K.-K., Macko, P., and Seltzer, M. Making a cloud provenance-aware. In *Proceedings of First workshop on on Theory and practice of provenance*. Berkeley, CA, USA, pp. 12:1–12:10, 2009.

Oliveira, D., Baião, F., and Mattoso, M. MiningFlow: Adding Semantics to Text Mining Workflows. In *Proceedings of Poster Session of The Brazilian Sysmposium on Databases*. João Pessoa, Brazil, pp. 15–18, 2008.

Taylor, I., Deelman, E., and Gannon, D., editors. *Workflows for e-Science*. Springer, 2007.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.* 39 (1): 50–55, 2009.