Captura de Metadados de Proveniência para *Workflows* Científicos em Nuvens Computacionais

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

Universidade Federal do Rio de Janeiro – Rio de Janeiro – RJ – Brasil

{kdu,danielc,serra,marta}@cos.ufrj.br, mluiza@nce.ufrj.br

Abstract. *Workflows are scientific abstractions used in the modeling of scientific experiments. High performance capabilities 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 workflow provenance. This paper presents an approach to support collecting metadata provenance of scientific experiments, based on an evolution of the Matrioshka architecture for the cloud environment. Preliminary results show that provenance metadata captured from the virtual components of the cloud can aid scientists to manage and reproduce their in silico experiments.*

Resumo. *Workflows científicos são abstrações utilizadas na modelagem de experimentos científicos. Eles muitas vezes demandam recursos de alto desempenho como clusters e grades computacionais. O modelo computacional chamado de computação em nuvem começa a ser adotado pela comunidade científica. No entanto, as nuvens computacionais ainda são incipientes no que se refere à coleta e registro de metadados de proveniência de workflows. Este artigo apresenta uma abordagem que apóia a coleta de metadados de proveniência de experimentos científicos, ela é baseada na evolução da arquitetura Matrioshka para o ambiente de nuvens. Os resultados preliminares apresentam os metadados já capturados a partir desse ambiente, com os quais, os cientistas podem gerenciar e reproduzir seus experimentos científicos in silico.*

1. Introdução

Experimentos em *e-Science* manipulam um crescente volume de dados (Hey *et al.*, 2009). Também chamados de experimentos *in silico* (Taylor *et al.*, 2007), estes experimentos se baseiam em modelos computacionais e podem ser executados em ambientes de computação de alto desempenho, tais como *clusters*, grades computacionais e mais recentemente, as nuvens computacionais. A computação em nuvem já vem sendo adotada no contexto da *e-Science* como um novo modelo de computação. As nuvens apresentam vantagens, principalmente no que se refere à elasticidade de recursos. Isto é, caso o cientista necessite de mais recursos, basta que o mesmo faça uma solicitação ao provedor da nuvem e os recursos serão disponibilizados. Assim, muitos cientistas já começaram a migrar seus experimentos para ambientes de nuvem (Hey *et al*., 2009).

Os experimentos *in silico*, são representados por meio do encadeamento de atividades, onde cada atividade é mapeada para uma aplicação, formando um fluxo coerente de

informações e controles, onde os dados de saída de um programa são entradas do próximo programa no fluxo. A esse encadeamento de atividades dá-se o nome de *workflow* científico (Taylor *et al.*, 2007). Os *workflows* científicos são gerenciados por Sistemas de Gerência de *Workflows* Científicos (SGWfC), que oferecem um arcabouço para executar, definir e monitorar as execuções dos *workflows* tanto local quanto remotamente. Existem diversos SGWfC, como por exemplo, o Kepler (Altintas *et al.,* 2006) e o VisTrails (Callahan *et al.*, 2006), cada um deles com características próprias. Entretanto, mesmo dentre os concebidos para operar em ambientes distribuídos, como o Pegasus (Deelman *et al.,* 2005), ainda não há apoio para gerência dos experimentos *in silico* executados em ambiente de nuvens.

Neste trabalho estamos interessados na fase de execução dos *workflows*. Para que um experimento não seja refutado pela comunidade científica, o mesmo deverá ser reproduzido sob as mesmas condições, mesmo que seja executado em ambientes distintos. Desta forma, os descritores associados ao *workflow*, como por exemplo, sua definição, os dados consumidos e os produzidos durante a sua execução são fundamentais para que o experimento seja considerado válido, consistente e ainda, capaz de ser reproduzido por terceiros (Cruz *et al.*, 2009). Esta categoria de descritores denomina-se metadados de proveniência (Freire *et al.*, 2008).

A literatura reporta vários sistemas capazes de capturar e gerenciar metadados de proveniência em ambientes distribuídos. Porém, grande parte deles é focada em ambientes como *clusters* e grades computacionais. Um exemplo destes mecanismos é a Matrioshka (Cruz *et al.*, 2008), cujo objetivo inicial era prover serviços que seriam acoplados ao SGWfC e implantados no ambiente distribuído para capturar e disponibilizar os metadados de proveniência desses ambientes. Entretanto, apesar de significar um avanço na área, a Matrioshka foi inicialmente projetada para *clusters* e grades computacionais, não contemplando as características dos ambientes de nuvem. Para realizar a captura de metadados de proveniência na nuvem é necessário levar em conta suas especificidades, a saber: os tipos de arquitetura de nuvem, virtualização de recursos e seus métodos de acesso, entre outras.

Este artigo propõe uma abordagem para o problema de captura de proveniência em ambientes de nuvem. Ele descreve a adaptação e utilização da arquitetura Matrioshka na captura de metadados de proveniência de *workflows* científicos executados em nuvens. Além disso, também apresenta um modelo de dados capaz de armazenar os metadados de proveniência específicos da nuvem e como estudo de caso um *workflow* de Mineração de Textos (MT) concebido e executado com o SGWfC VisTrails. Para alcançar este objetivo, foram realizadas modificações na arquitetura original da Matrioshka e foi também estendido o modelo de dados de proveniência*,* que considera a recomendação *Open Provenance Model* (OPM) (Moreau *et al.*, 2009), o qual propõe uma representação genérica de proveniência. O ambiente utilizado para a realização dos testes foi a nuvem da IBM (IBM, 2010), e os componentes da arquitetura foram desenvolvidos na linguagem Java.

Este artigo está organizado conforme a seguir. A Seção 2 faz um breve apanhado sobre os conceitos de computação em nuvem e proveniência e discute os trabalhos relacionados. A Seção 3 apresenta a arquitetura da Matrioshka adaptada para o contexto de nuvem. A Seção 4 relata um estudo de caso utilizando um *workflow* de MT. A Seção 5 conclui o artigo.

2. Computação em Nuvem e Proveniência

Foster *et al.* (2008) detalharam as diferenças principais entre grades computacionais e nuvens, definindo a computação em nuvem como "uma infraestrutura de computação, provida sob demanda, que oferece comunicação e controle, sendo servida a partir da rede, de forma compartilhada e dinamicamente escalável". Oliveira *et al.* (2010) classificam e descrevem as principais características dos ambientes de nuvem de acordo com uma perspectiva da *e-Science*. Uma das vantagens da nuvem para os experimentos é prover aos cientistas o acesso a uma grande variedade de recursos sem ter que necessariamente adquirir e configurar a infraestrutura computacional. São exemplos de experimentos *in silico* adaptados para nuvens, os projetos Sloan Digital Sky Survey e Berkley Water Center (Hey *et al*., 2009). Outra característica comum a muitos desses projetos é o uso intensivo de *workflows* científicos utilizando vários SGWfC. Consequentemente surge a necessidade da coleta de metadados de proveniência na nuvem, pois é necessário assegurar a reprodutibilidade desses experimentos. Sem esses metadados, o experimento tem sua avaliação e reprodução comprometidas. Por exemplo, em geral a execução em ambientes de nuvem ocorre de forma transparente para o cientista, ou seja, o que se passa na nuvem é uma "caixa preta". Portanto, é fundamental que os cientistas saibam quais os parâmetros foram utilizados e quais os produtos de dados foram gerados em cada execução. Entretanto, a captura e o gerenciamento de metadados de proveniência em ambientes distribuídos ainda representam uma questão em aberto (Freire *et al.*, 2008), (Mattoso *et al.*, 2010). Por exemplo, no ambiente de nuvem, quanto mais dados precisam ser trafegados utilizando a Internet, mais suscetível a falhas fica o sistema responsável por capturar e armazenar os dados de proveniência. Por esse motivo, a abordagem descrita neste artigo armazena os metadados de proveniência na própria nuvem, sendo recuperados para o ambiente local *a posteriori*.

Até o momento nenhum dos ambientes de nuvem oferecem, de forma nativa, meios capazes de capturar e armazenar metadados de proveniência produzidos por experimentos *in silico*. No entanto, já existem trabalhos que ressaltam a importância do tema como em Muniswamy-Reddy *et al.* (2009), onde são apresentadas algumas alternativas para o armazenamento da proveniência usando os serviços de computação em nuvem oferecidos pela Amazon EC2 (Amazon EC2, 2010) e utilizando o sistema PASS (Muniswamy-Reddy *et al.*, 2006). O PASS também é um sistema para armazenamento de proveniência distribuída, só que está intensamente associado à coleta de proveniência sobre os arquivos gerados, diferentemente da Matrioshka, que coleta metadados de proveniência sobre processos e informações do ambiente e também arquivos. O PASS propõe a utilização de três arquiteturas utilizando estruturas de armazenamento nativas da Amazon EC2, o Simple Storage Service (S3), o SimpleDB e o Simple Queueing Service (SQS), no entanto, não discute como e o que está sendo registrado no banco de dados de proveniência.

3. Matrioshka em Nuvens Computacionais

Esta seção descreve a abordagem que permite capturar e armazenar metadados de proveniência de *workflows* executados em nuvens. A proposta original da Matrioshka é focada em suprir algumas limitações existentes nos SGWfC em relação a coleta de metadados de proveniência distribuída. Ela atua como uma camada adicional e opera independentemente do SGWfC utilizado para executar o *workflow* científico, permitindo a coleta de metadados de proveniência nos ambientes distribuídos utilizados na execução de *workflows*. A nova abordagem visa minimizar a possibilidade da existência de silos de proveniência isolados, isto é, ela permite

que se reúnam os metadados de proveniência coletados dos ambientes de execução do *workflows* em um único modelo de dados.

3.1. Arquitetura Matrioshka

A arquitetura original foi concebida para ambientes de *clusters* e grades computacionais e operava acoplada ao SGWfC Kepler, por isso, utilizava serviços nativos desses ambientes, como por exemplo, escalonadores de processos, gerenciadores de fila, entre outros. Portanto, não apoiando características do ambiente de nuvens computacionais, como por exemplo, elasticidade de recursos, virtualização, independência de localização, entre outras. Para a Matrioshka ser utilizada nas nuvens, foi necessário adaptá-la para a infraestrutura oferecida pelos provedores de nuvens, já discutidas por Oliveira *et al.* (2010). A arquitetura original era composta por três componentes principais: *Provenance Broker*, *Provenance Eavesdrop* e o repositório de metadados de proveniência. Estes componentes operam no ambiente distribuído. No entanto, para que a Matrioshka opere satisfatoriamente no ambiente de nuvem houve a necessidade não só de alterar o modelo de dados, como também agregar novos componentes: o *Dispatcher* e o *Execution Broker*.

O *Provenance Broker* é o componente responsável por receber os metadados de proveniência, ou seja, aqueles relacionados com a execução das atividades do *workflow* no ambiente da nuvem, e armazená-los em um banco de dados. Quando a execução de alguma atividade do *workflow* ocorre na nuvem, os componentes *Provenance Broker* e *Provenance Eavesdrop* são invocados pelo componente *Dispatcher*. O *Provenance Broker* passa a receber os dados de proveniência capturados pelo *Provenance Eavesdrop*. Ao receber esses dados o *Provenance Broker* se encarrega de persisti-los em um modelo de dados armazenado na nuvem. Por sua vez, o componente *Provenance Eavesdrop* realiza a tarefa de coletar os dados de proveniência gerados pelas atividades e também produzidos pelo ambiente da nuvem. O *Provenance Broker* e o *Provenance Eavesdrop* são componentes remotos e trabalham com dados heterogêneos produzidos no ambiente de nuvem que podem ser originados de diversas fontes, como por exemplo, processos em execução, arquivos utilizados ou produzidos ou informações sobre as instâncias. O repositório de dados armazena os dados de proveniência associados à execução das atividades do *workflow* que compõem um experimento. O *Provenance Broker* e o *Provenance Eavesdrop* foram concebidos para serem independentes da infraestrutura de nuvem a ser utilizada, isto é, os componentes podem ser configurados nas mais diversas nuvens computacionais disponíveis.

O *Dispatcher* é um componente novo de execução local e deverá ser incluído na definição do *workflow* em um SGWfC. Este componente substitui uma ou mais atividades que inicialmente seriam executadas localmente. Ele é responsável pela invocação remota da execução de uma atividade numa instância da nuvem. É importante destacar que o programa invocado pela atividade remota deverá ser instalado *a priori* nas instâncias as quais o cientista possua acesso na nuvem. Através do *Dispatcher*, o cientista poderá configurar alguns parâmetros para acessar e utilizar as instâncias na nuvem, como por exemplo, seu *login*/senha, quantidade de instâncias a serem utilizadas, nome do programa a ser executado, dados e parâmetros de entrada do programa, entre outros. Esses parâmetros são armazenados em um arquivo de manifesto no formato XML. Esse arquivo contém especificações sobre as configurações de acesso às instâncias da nuvem. Ele possui a vantagem de ser tecnologicamente agnóstico do ponto de vista de sistemas operacionais. Além disso, ele também representa um conjunto de metadados de proveniência associada a execução de uma

atividade de um *workflow* na nuvem. O *Execution Broker* também é um novo componente necessário para a adaptação da Matrioshka à nuvem. Ele dispara a execução da atividade nas instâncias da nuvem que o cientista possui acesso e, quando a execução é concluída essa informação é repassada ao *Dispatcher*, para que o *workflow* prossiga com a execução de suas atividades locais. Em uma dada instância da nuvem são instalados o *Execution Broker*, o repositório de metadados e o *Provenance Broker*. Nas demais instâncias são instaladas várias cópias do *Provenance Eavesdrop* e os programas associados às atividades remotas do *workflow*. A Figura 1 é uma representação conceitual da arquitetura Matrioshka adaptada para o ambiente de nuvem. O *Provenance Eavesdrop* interage diretamente com as instâncias da nuvem, portanto, nesse componente e no repositório de metadados foram realizadas as principais adaptações necessárias para o funcionamento na nuvem. A troca de mensagens entre o ambiente local e entre as instâncias é realizada através de tunelamento seguro, promovido pelo protocolo SSH, onde são realizadas as transferências de dados de entrada e saída das atividades.

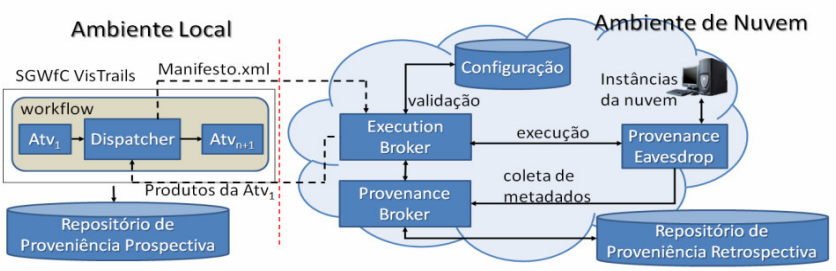

Figura 1. Matrioshka para ambiente de nuvem adaptada de Cruz *et al.* (2008).

3.2. Modelo de Dados de Proveniência

Matrioshka foi adaptada para o ambiente de nuvens com um novo modelo de metadados de proveniência motivado pelas diferenças dos dados disponíveis em *clusters* e grades computacionais. Por exemplo, as nuvens são fortemente baseadas nos conceitos de virtualização de recursos, onde um cientista pode acessar uma ou mais contas em diferentes provedores de serviço de nuvem que provêem diferentes tipos de instâncias no que se refere à configuração do *hardware* e *software*. Além disto, ao utilizarmos instâncias com configurações distintas, é necessário registrar em qual instância os produtos de dados das atividades do *workflow* se encontram, quais condições de processamento, quais os recursos consumidos, versão dos programas utilizados, dentre outros. O novo modelo de dados tem como base a mais recente recomendação do OPM em sua versão 1.1. Apesar de o OPM expressar as relações causais entre *Processos*, *Agentes*, *Artefatos* e *Papéis* existentes em *workflows,* o OPM é um modelo de referência não instanciável diretamente em um banco de dados. O OPM também visa facilitar a interoperabilidade de metadados de proveniência oriundos de ambientes heterogêneos de forma agnóstica do ponto de vista da tecnologia e dos SGWfC. Por esse motivo o OPM é considerado por diversos SGWfC (Moreau *et al.*, 2007) e (Freire *et al.*, 2008).

Na proposta original da Matrioshka havia descritores tais como, nós do *cluster* e detalhes do *jobs*, os quais eram recolhidos de forma a descrever o ambiente de execução. Além disso, não havia correlação com a atual versão do OPM. Entretanto, ao migrarmos para o ambiente de nuvem, muitos destes metadados perdem sentido. Por exemplo, o conceito de nó é substituído por máquinas virtuais (instâncias) disponibilizadas por provedores e cada instância pode ser diferente da outra no que se refere ao *hardware* e ao *software*. Além disto, ao utilizarmos instâncias diferentes, é necessário registrar em qual instância os produtos de dados

da atividade do *workflow* se encontram, versão dos programas utilizados, entre outras questões importantes para garantir a reprodutibilidade do experimento. A Figura 2 apresenta o modelo de dados (simplificado) de proveniência adotado pela Matrioshka para nuvens. Ele é representado como um diagrama de classes UML e, é resultado de um levantamento inicial sobre quais metadados de proveniência podem ser capturados pelos componentes *Provenance Eavesdrop e Provenance Broker*. O modelo de dados é composto por quatro partes principais: (i) elementos que representam os processos que serão distribuídos nas instâncias na nuvem, por exemplo, as atividades do *workflow*; (ii) elementos que representam os cientistas, associados a execução do *workflow*; (iii) elementos que representam os artefatos e recursos computacionais utilizados naquela execução do *workflow,* e por fim, (iv) elementos que representam informações relacionadas com a temporalidade do *workflow* e suas atividades.

Uma vez que o modelo de dados seguiu a recomendação do OPM, temos que, as classes *CloudOutput e CloudInstance* correspondem a representação conceitual de um *Artefato-OPM*, possuindo a mesma semântica, isto é, ambas representam estruturas digitais em sistemas de computação (*i.e.* parâmetros, bancos de dados, arquivos, instâncias, entre outros). A classe *CloudActivity* é mapeada como um *Processo-OPM*. Um processo representa uma ou mais ações que utilizam ou atuam sobre artefatos e que produzem novos artefatos. As classes *CloudProvider e CloudUser* representam um *Agente-OPM*. Um agente é o elemento que catalisa, possibilita, controla ou afeta a execução de um processo. As classes *CloudUserWorkflow*, *CloudUserInstance* e *CloudActivityInstance* são mapeados como *Papéis-OPM*. Um papel determina e correlaciona a função de um agente ou artefato em um processo. A classe *CloudExecution* representa o momento de execução de um processo na nuvem.

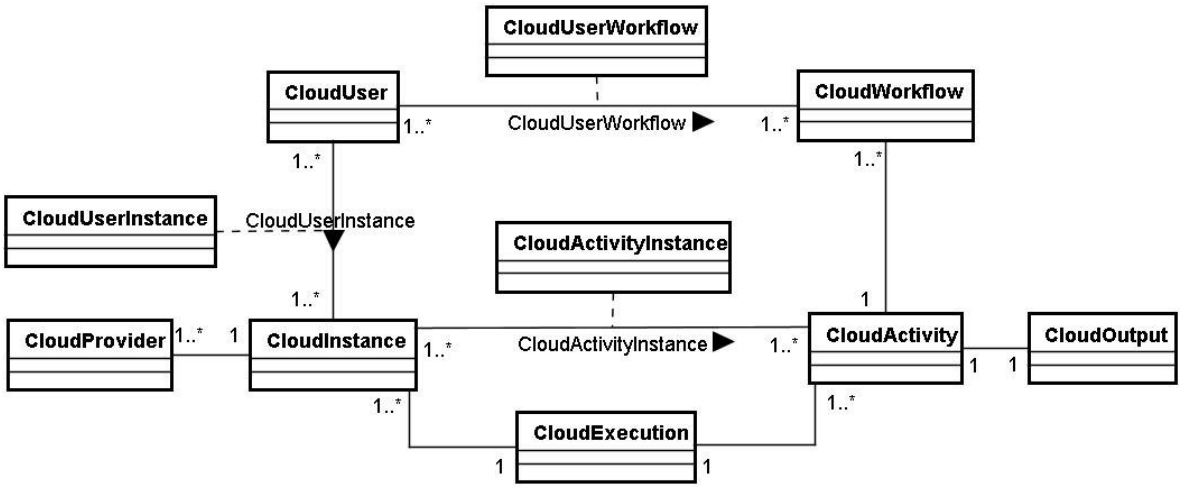

Figura 2. Modelo de Proveniência adaptado de Cruz *et al.* (2008).

4. Estudo de Caso

A MT é composta por três fases (pré-processamento, mineração e pós-processamento) e objetiva extrair conhecimento útil a partir de fontes de dados não estruturadas, como textos livres. A modelagem do processo de MT como *workflows* foi avaliada em (Oliveira *et al.*, 2007). Por ser este um primeiro estudo de viabilidade para a nuvem, o *workflow* utilizado atua apenas na fase de pré-processamento, isto é, preparando os dados textuais para serem minerados. O *workflow* foi modelado no SGWfC VisTrails (Figura 3) e executado na nuvem da IBM. A proveniência é coletada pelos componentes da Matrioshka e armazenados no SGBD DB2. O *workflow* é

composto por três atividades principais, a saber: limpeza dos dados (retirada de *stop words*, por exemplo), contagem de palavras e geração da tabela frequência de palavras × documento. Ao término das execuções é produzido um arquivo CSV para cada coleção contendo as freqüências de cada coleção pré-processada. Este arquivo é produzido nas instâncias e transferido *a posteriori* para a máquina do cientista.

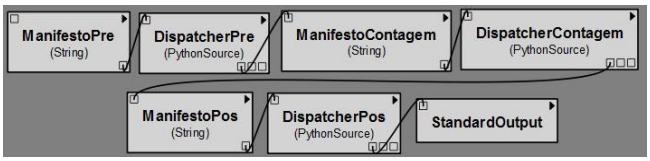

Figura 3. *Workflow* de MT especificado no SGWfC VisTrails.

Com este estudo de caso, pudemos executar um conjunto de testes relacionados à execução de *workflows* de MT em nuvem e capturar metadados de proveniência importantes, tais como, os identificadores de produtos de dados e a identificação das instâncias utilizadas (endereço IP). Além disso, foi possível identificar a instância e o tipo de BD que armazena os resultados do processamento. Por fim, também identificamos quais usuários executaram o *workflow*. Nesse estudo de caso foi utilizada uma coleção com aproximadamente 100 documentos no formato PDF. Esta coleção foi distribuída entre cinco instâncias para execução na nuvem da IBM. Em cada instância foram alocados 20 arquivos de entrada. A Tabela 1 apresenta parte do conjunto de metadados de proveniência que foram capturados pelos componentes da Matrioshka, observa-se que eles não poderiam ser coletados pelo SGWfC.

Tabela 1. Metadados de proveniência capturados na execução do *workflow* de MT na nuvem.

Arquivos produzidos	IP instâncias	Área de trabalho	IP BD
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)

Os endereços IP das instâncias (tanto de execução quanto do BD) são capturados na entidade *CloudInstance*. No caso das informações sobre os arquivos produzidos e área de trabalho, podem ser encontradas na entidade *CloudOutput*. Devido a restrições de espaço no artigo, nem todos os metadados capturados puderam ser explicados e exemplificados. Foram representados apenas os metadados mais significativos. Baseado nestes metadados é possível que o cientista saiba, por exemplo, em qual instância virtual estão os arquivos gerados pelo *workflow*, por exemplo, informação que os SGWfC locais não são capazes de fornecer sem a utilização da Matrioshka na nuvem.

5. Conclusão e Trabalhos Futuros

A computação em nuvem apresenta-se como uma interessante alternativa para a execução de experimentos baseados em *workflows* científicos que demandam ambientes computacionais distribuídos, principalmente devido à elasticidade de recursos e a alta disponibilidade. Porém, por ser uma tecnologia ainda incipiente em aplicações de *e-Science*, os SGWfC que oferecem apoio adequado à execução dos *workflows* em nuvens ainda é um problema em aberto, principalmente no que tange à captura e ao armazenamento de metadados de proveniência desse ambiente. Neste artigo, apresentamos uma arquitetura para coletar os metadados de

proveniência de *workflows* executados em ambientes de nuvens. Além disso, apresentamos um modelo de dados de proveniência que segue a recomendação OPM.

Apesar de ser um trabalho em andamento, os primeiros resultados são promissores, pois conseguiu-se capturar um conjunto inicial de dados que não poderiam ser coletados somente com os mecanismos locais de proveniência dos SGWfC. Em relação aos trabalhos futuros, faremos uma avaliação de escalabilidade da solução e também de desempenho a fim de verificar a sobrecarga da arquitetura. Além disso, novos estudos serão realizados visando a integração entre o modelo de dados apresentado com o modelo de proveniência local do SGWfC. Adicionalmente, serão executados novos *workflows* da área de bioinformática, em especial os de genômica comparativa, pois além de manipularem elevados volumes de dados, são de interesse dos nossos parceiros de pesquisa do Instituto Oswaldo Cruz.

Referências

Altintas, I., Ludaescher, B., Klasky, S., Vouk, M., (2006), "Introduction to scientific *workflow* management and the Kepler system". ACM/IEEE *Super Computing 06*, Tampa, FL.

Amazon EC2, (2010). Amazon Elastic Compute Cloud (Amazon EC2). *Amazon Elastic Compute Cloud (Amazon EC2)*. Disponível em: http://aws.amazon.com/ec2/. Acesso em: 15 Jun 2010.

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., Vo, H. T., (2006), "VisTrails: visualization meets data management". In: *Proc. SIGMOD 2006*, p. 745-747, USA.

Cruz, S. M. S. D., Barros, P. M., Bisch, P. M., Campos, M. L. M., Mattoso, M., (2008), "Provenance Services for Distributed *Workflows*". In: *Proc. CCGrid2008* p. 526-533.

Cruz, S.M.S., Campos, M. L. M., Mattoso, M. (2009) "Towards a Taxonomy of Provenance in Scientific *Workflow* Management Systems" In: IEEE-Services´09 pp. 259-266.

Deelman, E. et al, Pegasus (2005) a Framework for Mapping Complex Scientific *Workflows* onto Distributed Systems. Scientific Programming Journal, v.13(3), p.219-237.

Foster, I., Zhao, Y., Raicu, I., Lu, S., (2008), "Cloud Computing and Grid Computing 360-Degree Compared". In: *Grid Computing Environments Workshop, 2008. GCE '08*, p. 10, 1

Freire, J., Koop, D., Santos, E., Silva, C. T., (2008), "Provenance for Computational Tasks: A Survey", *Computing in Science and Engineering, v.10*, n. 3, p. 11-21.

Hey, T., Tansley, S., Tolle ., (2009), "The Fourth Paradigm: Data-Intensive Scientific Discovery" Microsoft Research, 284pp.

IBM, (2010). IBM Smart Business Development & Test - Home. Disponível em: https://www-949.ibm.com/cloud/developer/dashboard. Acesso em: 15 Jun 2010.

Mattoso, M., Werner, C., Travassos, G., Braganholo, V., Murta, L., Ogasawara, E., Oliveira, D., Cruz, S., Martinho, W. (2010),"Towards Supporting the Life Cycle of Large Scale Scientific Experiments", IJBPIM, v. 5, p. 79-92, 2010.

Moreau, L. Freire, J. Myers, J., Futrelle, J et al. (2009), "The Open Provenance Model - Core Specification v1.1", *A ser publicado Future Generation Computer Science.*

Muniswamy-Reddy, K., Holland, D., Braun, U., Seltzer, M., (2006), "Provenance-aware storage systems". In: *Proceedings of the 2006 USENIX Annual Technical Conference*.

Muniswamy-Reddy, K., Macko, P., Seltzer, M., (2009), "Making a cloud provenance-aware". In: *1st workshop on on Theory and practice of provenance*, p. 1-10, San Francisco, CA.

Oliveira, D., Baião, F., Mattoso, M., (2010), "Towards a Taxonomy for Cloud Computing from an e-Science Perspective", *A ser publicado: Cloud Computing: principles, applications and architecture.* Springer.

Oliveira, D. C. M.; Baiao, F.; Mattoso, M. (2007). MiningFlow: Adding Semantics to Text Mining *Workflows*. In: Sessão de Pôsteres do Simpósio Brasileiro de Banco de Dados, João Pessoa

Taylor, I. J., Deelman, E., Gannon, D. B., Shields, M., (Eds.), (2007), *Workflows for e-Science: Scientific Workflows for Grids*. 1 ed. Springer.