Distribuição de Bases de Dados de Proveniência na Nuvem *

Edimar Santos[1], Vanessa Assis[1], Flavio Costa[1], Daniel de Oliveira[2] e Marta Mattoso[1]

[1]Programa de Engenharia de Sistemas e Computação – COPPE / UFRJ
[2]Instituto de Computação - Universidade Federal Fluminense - UFF
{ebabilon,vmarques,flscosta,marta}@cos.ufrj.br, danielcmo@ic.uff.br

Resumo. Dados de proveniência no contexto de *workflows* científicos são peças fundamentais, pois, por meio deles, os experimentos são passíveis de reprodução e validação. O histórico da execução dos *workflows* é fundamental também para a gerência da execução de novos *workflows* uma vez que possibilitam às máquinas de *workflow* realizar predições sobre desempenho ou custo financeiro de nuvens de computadores. *Workflows,* com dados em larga escala, executados em nuvens, são com frequência alocados em máquinas virtuais distribuídas fisicamente. As soluções existentes coletam os dados de proveniência de forma distribuída e os armazenam de modo centralizado em único repositório, após o término da execução do workflow. Além da capacidade de reprodução, dados de proveniência permitem um acompanhamento refinado por parte do cientista, quando disponibilizados à medida que são gerados, durante a execução do workflow. Porém, quando os dados de proveniência só estão disponíveis para consulta após a execução do workflow, seu uso fica limitado. Para permitir consultas durante a execução do workflow, o acesso ao banco de dados de proveniência deve estar em sintonia com a máquina de execução distribuída de workflows. Este artigo discute aspectos de projeto de distribuição de dados de proveniência, levando em consideração o esquema de representação de proveniência do W3C, aspectos de processamento distribuído de consultas em nuvens de computadores e considerando a execução distribuída do workflow. A estratégia aqui adotada trouxe melhoria de desempenho para as consultas que submetemos em tempo de execução dos *workflows* aumentando assim a eficiência dos *workflows* científicos testados.

Palavras-chave: *Workflow*, Proveniência de dados, projeto de distribuição

1. INTRODUÇÃO

Os experimentos científicos baseados em simulações computacionais são comumente realizados por múltiplas combinações de programas, onde cada um deles possui um conjunto de parâmetros e dados de entrada. Esses experimentos podem ser modelados como *workflows* científicos (Deelman *et al.* 2009). Um *workflow* científico é uma abstração que representa o encadeamento de programas formando um fluxo coerente que produz um resultado final. Tais *workflows* são modelados e executados para validar hipóteses levantadas pelos cientistas, e por muitas vezes são complexos demandando a utilização de recursos computacionais de capacidade de alto desempenho. Como os experimentos científicos necessitam ter sua reprodução garantida, a utilização da proveniência se torna uma questão fundamental (Freire *et al.* 2008). Os dados relativos à execução do *workflow* devem ser coletados e passíveis de consulta. Essa gerência dos dados de proveniência se torna mais complexa quando os *workflows* precisam ser executados em paralelo em ambientes de alto desempenho, como as nuvens de computadores (Vaquero *et al.* 2009). É fundamental que a proveniência registre qual atividade gerou qual dado e onde esse dado está armazenado.

Nas máquinas de *workflows* atuais, os dados de proveniência são coletados durante a execução do *workflow*, mas não são disponibilizados para consulta enquanto o *workflow* não termina de executar. A disponibilização dos dados de proveniência para consulta, durante a execução do *workflow*, pode prover uma série de vantagens para o cientista. Com a proveniência consultada em tempo real, é possível realizar o escalonamento adaptativo de *workflows* (Oliveira *et al.* 2011), a supervisão da execução do *workflow* (*i.e. workflow steering*) (Dias *et al.* 2011) e a gerência de falhas das atividades (Costa *et al.* 2012).

* Esse trabalho foi financiado parcialmente pela Capes, Faperj e CNPq

Entretanto, a literatura sobre sistemas e máquinas de *workflows* não apresenta soluções para consultas a dados de proveniência distribuídos durante a execução do workflow. Os repositórios de proveniência são disponibilizados somente após as execuções do workflow, de forma centralizada, ou seja, armazenam em uma mesma base todos os dados das execuções de todos os *workflows* independentemente de onde e por quem foram executados. Este cenário torna-se ainda mais complexo quando há equipes geograficamente dispersas trabalhando nos mesmos *workflows* cuja análise dos resultados é colaborativa e em tempo real.

Alguns poucos trabalhos tratam da distribuição de dados de proveniência. Allen *et al.* 2011, Freire *et al.* 2008, Zhou *et al.* 2011. Nenhum destes trabalhos tem o objetivo de disponibilizar dados de proveniência durante a execução distribuída do workflow. Entretanto, alguns desses trabalhos possuem um grau de semelhança com a proposta apresentada neste artigo. O artigo de Zhou *et al.* (2011) apresenta uma abordagem de proveniência para sistemas distribuídos. Entretanto, Zhou *et al.* fazem justamente o contrário do que é proposto aqui. Os autores assumem que os dados estão distribuídos em diversos sítios e realizam consolidação para que se possa analisar o desempenho de execuções dinâmicas no sistema como um todo. No trabalho de Allen *et al.* (2011) é apresentada uma arquitetura para o compartilhamento de proveniência, mas assume que os dados já estão distribuídos. A arquitetura proposta poderia ser utilizada como complemento à estratégia de fragmentação que é analisada no presente artigo.

O problema de distribuição de dados já foi amplamente discutido e diversas soluções são apresentadas por Özsu e Valduriez (2011). Entretanto, pouco se sabe sobre o comportamento de bancos de dados distribuídos em nuvens de computadores, foco de deste trabalho, por se tratar do ambiente utilizado por muitos grupos de pesquisa para execução de seus experimentos. A complexidade se torna maior quando o esquema de fragmentação precisa estar de acordo com um padrão, no caso o W3C, e acompanhar o dinamismo da máquina distribuída de execução de workflows em nuvens.

O objetivo deste artigo é avaliar o comportamento de consultas a dados de proveniência que são gerados em tempo real, de modo distribuído, por máquinas de execução de *workflows* em nuvens, de modo a propor um projeto de distribuição que atenda ao cenário apresentado. Ao trabalharmos com distribuição de dados na nuvem, diferente de outros ambientes de computação paralela, a transferência de dados entre os nós é um ponto que deve ser sempre investigado. Este artigo apresenta uma proposta de execução distribuída de consultas sobre uma fragmentação de dados de proveniência e discute o comportamento de consultas submetidas durante uma execução real de workflow na nuvem. Para tal, partiu-se do esquema de dados de proveniência utilizado pela máquina de execução distribuída de *workflows* SciCumulus (Oliveira *et al.* 2011). O esquema do SciCumulus segue o PROV-Wf (Costa *et al.* 2013) que estende o modelo PROV do W3C para a representação de proveniência em *workflows*.

Além da introdução, este artigo possui 3 seções. A Seção 2 apresenta o projeto de distribuição dos dados de proveniência. A Seção 3 avalia o processamento distribuído das consultas e a Seção 4 conclui.

2. PROJETO DE DISTRIBUIÇÃO E CONTROLE DE ACESSO AOS DADOS DE PROVENIÊNCIA

Um projeto de distribuição realiza duas decisões, a fragmentação e a alocação (Özsu e Valduriez 2011). Além disso, podemos destacar também o controle de acesso aos fragmentos. Para o projeto de fragmentação, são analisadas as consultas mais frequentes e a alocação determina onde os fragmentos e eventuais réplicas serão alocados. Durante a execução do workflow, o cientista deverá analisar apenas as tuplas associadas ao *workflow* que ele está acompanhando, assim, a fragmentação horizontal, por *workflow*, se aplica de forma adequada para esse cenário. Por outro lado, a proveniência é usada pela máquina de execução do *workflow* para funções como o escalonamento de uma determinada atividade. Nesse caso, apenas os tempos de execução da atividade e os indicativos de falha são suficientes. Essas consultas à base de proveniência acessam poucos atributos, mas necessitam de todas as tuplas, já que irão calcular a média de tempo de execução, desvio-padrão, etc. Neste cenário, a fragmentação vertical parece ser a mais adequada, mesmo porque, somente os tempos estariam acessíveis, sem apresentar os resultados da pesquisa que podem estar armazenados no mesmo esquema de proveniência.

Seja o seguinte cenário (com nomes propositalmente fictícios, mas com um *workflow* real): Pedro é um cientista, localizado em São Paulo, que executa em seus experimentos o *workflow* SciPhy (Ocaña *et al.* 2011), que realiza análises filogenéticas. Em sua base de proveniência, já existe uma grande quantidade de informações referentes a execuções do SciPhy. Um colega de Pedro que reside em Dublin, chamado Robert, está se familiarizando com o *workflow* SciPhy e pretende utilizá-lo para experimentos de análise filogenética em um futuro próximo. Assim, Pedro pretende executar experimentos em conjunto com Robert de forma que possam realizar uma análise colaborativa. Entretanto, existem outros cientistas executando o *workflow SciPhy* que Pedro não deseja compartilhar informação, como Dr. Kobayashi do Japão. Robert deve poder usar os dados de Pedro para que sua máquina de *workflow* possa realizar um escalonamento com base no histórico ou possa descobrir pontos de falha de maneira eficiente. Esse é um cenário comum em que Pedro pode compartilhar seus dados com Robert sem que terceiros possam acessá-los, e assim Robert possa obter esses dados de proveniência da maneira mais eficiente possível.

No cenário apresentado, pode-se fragmentar horizontalmente a base de dados de proveniência de acordo com o nome do *workflow* em execução, pois essa é uma distribuição natural, na qual os fragmentos representam os conjuntos de *workflows* executados em uma determinada localidade. Nesse caso, Robert faria sua consulta apenas no fragmento que contém as informações das execuções do *workflow* SciPhy executado em Sao Paulo. Posteriormente, usaria seu próprio sítio para armazenar seus dados, mas continuaria utilizando a massa de dados do fragmento de São Paulo para compor suas análises. Se um mesmo *workflow* pode ser executado tanto na Irlanda quanto em Tóquio, ou mesmo no Brasil como no exemplo apresentado, a próxima decisão a ser tomada seria de alocação com replicação dos dados relativos ao *workflow*.

Na fragmentação horizontal, os fragmentos possuem todos os atributos da relação original, mas há uma redução na quantidade de tuplas a serem armazenadas e consultadas, o que favorece a diminuição do tempo de resposta. Considerando que os atributos mais frequentes para as consultas, são o nome do *workflow* e a sua localidade, e tendo em vista que o cientista tende a fazer um número maior de consultas em experimentos executados por seu grupo de pesquisas, a escolha por esses atributos para a fragmentação horizontal também acaba se tornando uma decisão natural. Assim, os atributos utilizados no projeto de fragmentação foram *tag* (nome do *workflow* que se encontra na classe *Workflow* do Prov-Wf) e localidade. Tal projeto só foi possível devido à estratégia de se adotar um modelo único derivado do PROV. Aplicou-se então o algoritmo de fragmentação horizontal baseado em mintermos, conforme em (Özsu e Valduriez 2011). Os predicados simples utilizados são:

P1. *tag*="SciPhy"; P2. *tag*="SciPhylomics"; P3. *tag*="SciEvol"; P4. *tag*="ExpX"; P5. *tag*="exp"; P6. *tag*="Scimult"; P7. localidade="Tóquio"; P8. localidade="Dublin"; P9. localidade="São Paulo".

Foram criados os fragmentos correspondentes a cada um dos mintermos finais para os *workflows* com as combinações das *tags* "SciPhy", "SciPhylomics", "SciEvol", "ExpX", "Exp" e "Scimult" para cada uma das localidades "Dublin", "São Paulo" e "Tóquio". Além disso, para garantir a completude, existe também um fragmento chamado *wk_outros*, para cada uma das localidades, onde devem ser armazenados *workflows* com *tags* diferentes das apresentadas anteriormente que foram elencadas a partir das consultas mais frequentes. Um total de 21 fragmentos foram gerados: sete fragmentos chamados *wk_sciphy, wk_sciphylomics, wk_scievol, wk_expx, wk_exp, wk_simultaneous* e *wk_outros* para cada uma das três localidades Dublin, São Paulo e Tóquio.

Além disso, criamos um repositório central com todos os fragmentos em questão, de modo que para todo fragmento criado, existe uma réplica em Singapura que possui os dados relativos a todo o conjunto de *workflows* executados. Os alvos de alocação foram então os três sítios: São Paulo, Dublin e Tóquio (utilizamos o serviço da Amazon AWS que permite criar máquinas virtuais especificando a localização física de servidores em várias partes do mundo). Para aumentar a disponibilidade dos dados, todos os fragmentos foram enviados para Singapura onde formaram um repositório centralizado sendo replicados após o projeto de distribuição, aumentando a disponibilidade em caso de falhas.

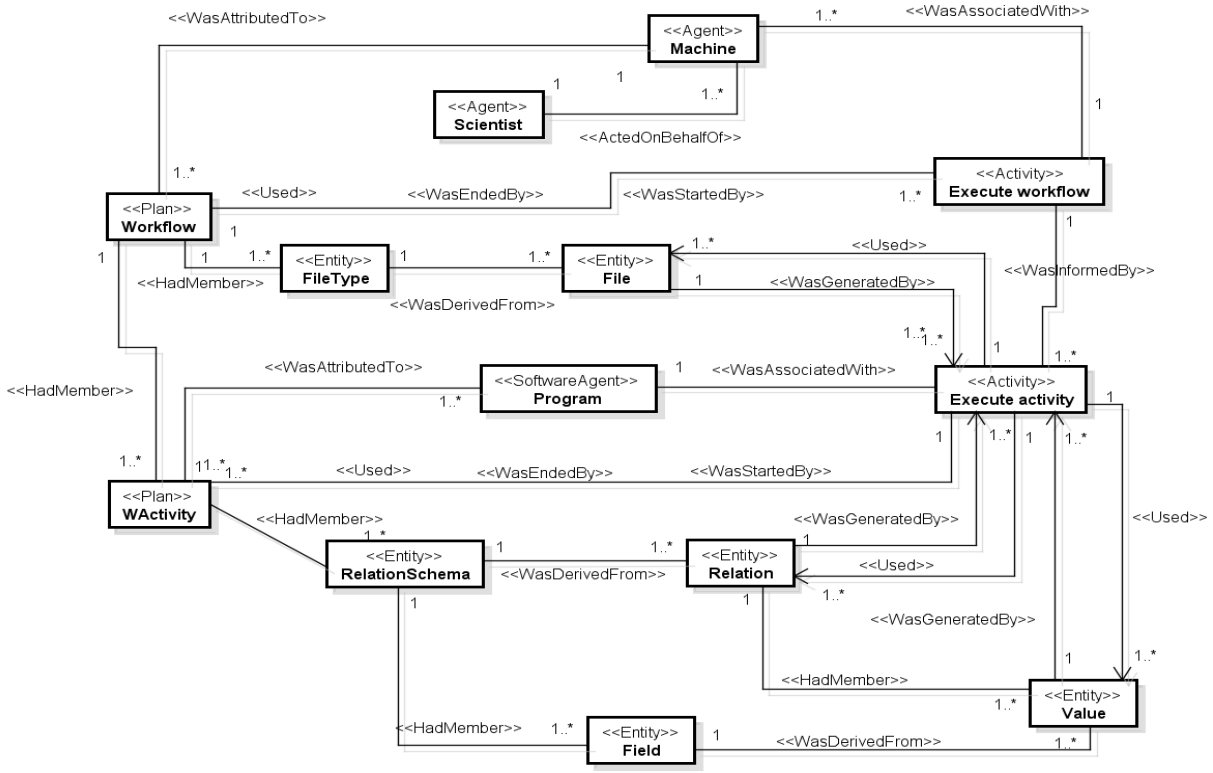

Figura 1 O esquema de proveniência PROV-Wf como proposto por Costa *et al.* (2013)

Como estamos trabalhando com fragmentos de dados (mesmo no caso de Singapura onde esses fragmentos são agrupados) foi possível criar um controle de acesso em nível de fragmento, ou seja, para cada equipe de pesquisa, eu tenho um conjunto de fragmentos disponíveis. Esses fragmentos podem estar localizados no sítio da equipe, no sítio de uma outra equipe colaboradora ou no sítio central em Singapura.

3. AVALIAÇÃO DE CONSULTAS SOBRE DADOS DE PROVENIÊNCIA FRAGMENTADOS

Para realizar uma análise de desempenho de consultas sobre os fragmentos gerados, foram consideradas consultas que são frequentemente utilizadas pelos cientistas dos workflows de análise filogenética. Em nossos experimentos, os dados da base de proveniência vão sendo disponibilizados para consulta, à medida que o workflow vai sendo executados. Utilizamos então algumas consultas que são submetidas durante a execução do workflow (sendo que nestes casos, para que os resultados fossem sempre os mesmos e fosse possível fazer a média de 10 execuções, as consultas foram submetidas de modo desacoplado da execução, sem prejuízo à análise dos resultados). Para a avaliação, foram selecionadas nove consultas consideradas frequentes, descritas a seguir: (1) Listar os rótulos e as descrições de todas as atividades de um determinado *workflow*; (2) Listar todas as execuções de cada uma das atividades de um determinado *workflow;* (3) Listar todas as atividades com estado diferente de "*finished*" de um determinado *workflow* em São Paulo, ou seja, todas as atividades que ainda não foram executadas ou não finalizaram sua execução; (4) Listar a duração de todas as atividades com estado igual a "*finished*" de um determinado *workflow* de Dublin; (5) Listar o horário de início de cada uma das atividades com status igual a "*running*" de um determinado *workflow* em Tóquio; (6) Buscar a duração de cada uma das execuções em workflows que contêm uma determinada atividade; (7) Buscar a mensagem de erro, obtida com o atributo *terr*, de todas as execuções de atividades com status de saída diferente de zero (*i.e.* sem erro) de um determinado *workflow*; (8) Listar todos os *workflows* que contêm uma determinada atividade especificada e (9) Listar, por ordem crescente de execuções de *workflows*, as datas e horas de início e término das ativações, *tags*

dos *workflows* e suas descrições, bem como o nome de todas as atividades associadas a todas as execuções dos *workflows* que foram executados sem erro. Cada uma das nove consultas apresentadas foi executada no ambiente distribuído e no ambiente centralizado, ambos como máquinas virtuais na nuvem da Amazon. O ambiente distribuído não usou a base centralizada para as consultas, assim, quando a consulta envolvia fragmentos não locais, consultas remotas eram submetidas nas máquinas virtuais correspondentes e os resultados consolidados localmente. Os tempos de resposta, obtidos com a execução das consultas (Tabela 1), foram comparados a fim de analisar o desempenho atingido utilizando o projeto de distribuição apresentado e como essa estratégia se comporta em relação ao ambiente centralizado em termos de tempo de resposta das consultas no ambiente de nuvens computacionais.

Tabela 1. Resultados das consultas sobre a base fragmentada e centralizada

Consulta	Tuplas acessadas	Tempo em ms (centralizado)	Tempo em ms (distribuído)
C1	105	826	405
C2	5.828	8.408	1.852
C3	15	391	208
C4	23	374	209
C5	11	406	199
C6	1.614	1.278	1.352
C7	125	1.743	773
C8	42	231	722
C9	15.259	16.973	26.052

Para a realização dessa análise de desempenho foram efetuadas cerca de dez execuções de cada uma das consultas descritas anteriormente, obtendo assim uma média dos tempos de execução. Por terem apresentado um desvio-padrão desprezível, as variações não são apresentadas. A escolha das consultas levou em consideração que a maioria dos cientistas realiza pesquisas buscando um *workflow*, em particular, o que ele está executando naquele momento. Podemos observar, conforme o esperado, que ocorreu um ganho significativo nos tempos de execução para as consultas realizadas sobre fragmentos específicos locais, como foi o caso das consultas C3, C4 e C5. Esse resultado era esperado, pois essas consultas utilizam como restrição a pesquisa por um determinado *workflow* e sua localidade, assim acessam um só fragmento local. Entretanto, mesmo em consultas com acesso remoto, mas direcionadas a fragmentos específicos, houve ganho significativo, como foi o caso de C1, C2 e C7. Cabe ressaltar que C2 acessa um número significativo de tuplas e obteve uma redução também importante, C2 sobre a base centralizada foi cerca de 4 vezes mais lenta que a distribuída. Esse resultado mostra que o paralelismo em consultas frequentes desse tipo podem compensar perdas em outros casos como C8 e C9. Cabe observar que se as consultas 8 e 9 forem as mais frequentes, uma fragmentação não seria recomendada e talvez uma replicação total da base nos sítios mais ativos poderia ser considerada. Podemos citar também o caso particular de C6, que não envolve a pesquisa por um *workflow* específico, e que busca a duração das execuções de uma determinada atividade. O tempo de resposta para essa pesquisa apresentou resultados equivalentes na base centralizada e distribuída, isso ocorre devido ao acesso específico à tabela atividade em cada sítio distribuído em paralelo. Ou seja, o processamento em paralelo sobre os fragmentos distribuídos acabou por tornar a execução distribuída, competitiva com a centralizada. As consultas que requerem percorrer todos os sítios acabam sendo penalizadas pelo tempo de comunicação e transferência de dados, levando em consideração a distância física entre os sítios. É importante lembrar que esse tipo de influência da distância reflete a realidade, tendo em vista que os centros de pesquisa estão espalhados pelo mundo, porém, considerando um conjunto de consultas analíticas, espera-se que os ganhos superem as perdas, ou seja, bastam duas execuções da C2 para compensar uma execução da C9. Por outro lado, se considerarmos um projeto de replicação que contemple um sítio contendo todos os fragmentos, como o de Singapura, o processamento distribuído de consultas poderia direcionar essa classe de consultas diretamente à base centralizada. Como as alterações da base de proveniência é baseada em inserções sem atualização de tuplas, o custo de manutenção da consistência não é muito significativo.

4. CONCLUSÃO

Os *workflows* científicos apresentam-se como uma abordagem fundamental para a modelagem de experimentos científicos baseados em simulações. A gerência da proveniência desses *workflows* é uma questão fundamental, pois além da análise do experimento, auxilia a reprodução, peça fundamental do processo científico. Além da reprodução, os dados de proveniência podem ajudar a máquina de execução do workflow no escalonamento, detecção de falhas, dimensionamento do ambiente, etc., pois contêm informações acerca das atividades executadas, as diferentes ativações utilizadas, os tempos de execução, os resultados gerados, possíveis erros ocorridos dentre outros. Obter essas informações traz grandes benefícios para os cientistas, pois a partir delas é possível abortar uma execução que não caminha na direção almejada e, em caso de sucesso, reproduzir a execução de um *workflow*, além de realizar um melhor gerenciamento dos recursos necessários e análises de desempenho, bem como detectar erros com maior facilidade e em tempo real. Como os dados de proveniência são utilizados frequentemente para tomar decisões em tempo real, é importante ter consultas com um tempo de resposta pequeno, para que essas não afetem o desempenho do *workflow*. Entretanto, as máquinas de *workflows* de hoje utilizam repositórios de proveniência centralizados, o que impõe pontos de falha e problemas de segurança e desempenho. Tendo isso em mente, foi proposto neste artigo uma estratégia de fragmentação e alocação dos dados de proveniência de maneira a minimizar o tempo necessário para que as consultas sejam realizadas. Essa estratégia levou em consideração dois fatores principais: o fato de que as consultas realizadas por um cientista em geral giram em torno do *workflow* em que ele está trabalhando; e que, no geral, os dados gerados por aquele cientista são armazenados nos sítios mais próximos a ele. Assim, foi realizada uma fragmentação horizontal sobre os atributos *tag* e localidade, que representam respectivamente o rótulo de identificação do *workflow* e o local onde o mesmo foi executado. Uma série de consultas, frequentemente utilizadas por cientistas, foi selecionada e avaliada tanto no ambiente distribuído, como no ambiente centralizado. Os resultados obtidos indicam uma melhora significativa no ambiente distribuído em consultas realizadas em um único fragmento, mostrando ganhos significativos sobre o tempo de execução da consulta na base centralizada mesmo considerando sítios distantes fisicamente. Como trabalho futuro será realizado uma análise da fragmentação vertical e híbrida.

REFERENCES

Allen, M., Chapman, A., Blaustein, B., Seligman, L., (2011), "Getting It Together: Enabling Multi-organization Provenance Exchange". In: *TaPP 2011*, Athens, Greece.

Costa, F., Oliveira, D., Ocaña, K., Ogasawara, E., Mattoso, M., (2012), "Enabling Re-Executions of Parallel Scientific Workflows Using Runtime Provenance Data. In: 4th International Provenance and Annotation Workshop"

Costa, F., Silva, V., Oliveira, D., Ocaña, K., Dias, J., Ogasawara, E., Mattoso, M., (2013), "Capturing and Querying Workflow Runtime Provenance with PROV: a Practical Approach". In: *BigProv'13*, Genova, Italy.

Deelman, E., Gannon, D., Shields, M., Taylor, I., (2009), "Workflows and e-Science: An overview of workflow system features and capabilities", *Future Generation Computer Systems*, v. 25, n. 5, p. 528–540.

Dias, J., Ogasawara, E., Oliveira, D., Porto, F., Coutinho, A., Mattoso, M., (2011), "Supporting Dynamic Parameter Sweep in Adaptive and User-Steered Workflow". In: *6th WORKS*, p. 31–36, Seattle, WA, USA.

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.

Missier, P., Belhajjame, K., Cheney, J., (2013), "The W3C PROV family of specifications for modelling provenance metadata". In: *Proceedings of the 16th EDBT*, p. 773–776, New York, NY, USA.

Ocaña, K. A. C. S., Oliveira, D., Ogasawara, E., Dávila, A. M. R., Lima, A. A. B., Mattoso, M., (2011), "SciPhy: A Cloud-Based Workflow for Phylogenetic Analysis of Drug Targets in Protozoan Genomes", BSB 2011: Springer, p. 66–70.

Oliveira, D., Ogasawara, E., Ocaña, K., Baião, F., Mattoso, M., (2011), "An Adaptive Parallel Execution Strategy for Cloud-based Scientific Workflows", *Concurrency and Computation: Practice and Experience*, v. (online)

Özsu, M. T., Valduriez, P., (2011), *Principles of Distributed Database Systems*. 3 ed. New York, Springer.

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

Zhou, W., Ding, L., Haerberlen, A., Ives, Z. G., Loo, B. T., (2011), "TAP: Time-aware Provenance for Distributed Systems". In: *Proc. of TaPP'11*, Greece.