Reprodução de Experimentos Científicos Usando Nuvens

Ary Henrique M. de Oliveira[1,2], Murilo de Souza Martins[1], Igor Modesto[1], Daniel de Oliveira[2], Marta Mattoso[2]

[1] Universidade Federal do Tocantins, Palmas-TO, Brasil

[2] COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro-RJ, Brasil

{aryhenrique, murilosm, igorma}@uft.edu.br, {danielc, marta}@cos.ufrj.br

Resumo. *Workflows* científicos são utilizados para modelar experimentos computacionais. Os resultados desses experimentos são publicados e compartilhados na forma de artigos publicados em veículos científicos. Entretanto, para que tais resultados sejam cientificamente válidos eles devem ser passíveis de reprodução. Pesquisadores da área de *e-Science* têm a necessidade de compartilhar os artefatos utilizados para a geração dos resultados, dentre eles, os dados de entrada do *workflow* e os parâmetros utilizados no experimento. Entretanto, reproduzir um experimento baseado nestes artefatos não é uma tarefa trivial. Apesar de o *workflow* especificar o protocolo de execução, com dados e parâmetros de entrada disponíveis, nem sempre o ambiente de execução está acessível. Programas que foram originalmente utilizados podem estar obsoletos, versões de bibliotecas podem não ser mais compatíveis além de ambientes que podem não estar mais disponíveis para o cientista. Esse cenário se torna ainda mais complexo quando tratamos de reproduções de longo prazo, como por exemplo, diversos anos após a execução que levou aos resultados publicados. Diante deste problema, este artigo propõe uma abordagem desenvolvida na forma de um componente de *software* chamada ReproeScience para reprodução do ambiente onde o experimento computacional foi originalmente executado, de forma que o mesmo possa ser instanciado sob demanda e reproduzido em iguais condições. Para isto é proposta a utilização da tecnologia de máquinas virtuais de nuvens de computadores como arcabouço básico.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Databases

1. INTRODUÇÃO

Durante as últimas décadas, a computação se estabeleceu como o quarto pilar da ciência para apoiar a formulação de teorias, uma vez que muitos experimentos podem se beneficiar de simulações executadas em ambientes computacionais (Taylor *et al.* 2007). Alguns dos problemas teóricos, que são analiticamente complexos de serem solucionados, podem ser resolvidos por meio de simulações e da utilização de recursos computacionais. Cada simulação desta categoria é caracterizada pelo encadeamento de programas em um fluxo coerente de atividades. A execução de experimentos baseados em simulação é apoiada por diversas técnicas e abordagens, como os *workflows* científicos. Um *workflow* científico é uma abstração que define as etapas de execução de um experimento e a sequência em que tais etapas ocorrem, de forma a corroborar ou refutar uma hipótese científica (Jarrard 2001). Cada etapa (*i.e.* atividade do *workflow*) envolve a execução de um determinado programa, que é responsável pela transformação dos dados de entrada e a produção de dados de saída. Os *workflows* científicos são modelados, executados e monitorados pelos chamados Sistemas de Gerência de *Workflows* Científicos (SGWfC).

A maioria desses *workflows* é computacionalmente intensiva e demanda ambientes de alto desempenho (Mattoso *et al.* 2010) em sua execução. Desta forma, é uma necessidade real a utilização de ambientes distribuídos, de grande capacidade de processamento e muitas vezes heterogêneos, aliados à aplicação de técnicas de paralelismo. Como exemplo destes ambientes, podemos citar os *clusters*, as grades, e mais recentemente as nuvens de computadores (Vaquero *et al.* 2009).

Para que um experimento seja válido sob o ponto de vista científico, o seu resultado deve ser passível de reprodução por terceiros. No cenário científico, se um cientista publica seus resultados de

* Este artigo foi parcialmente financiado pelo CNPq, CAPES e FAPERJ

uma forma que somente ele possa reproduzi-los, a comunidade pode considerar sua pesquisa inválida. Para atingir tal nível de reprodução, necessita-se possuir dados descritores tanto do ambiente de execução quanto dos experimentos ou *workflows* propriamente ditos. Estes dados podem ser obtidos a partir de dados de proveniência (Freire *et al.* 2008). Dados de proveniência são automaticamente coletados por SGWfC ao longo da execução do *workflow* e são fundamentais para reproduzir e validar uma determinada execução do *workflow*. Apesar de já existirem diversas abordagens para a captura dos dados de proveniência, seja em ambientes sequenciais ou distribuídos (Altintas *et al.* 2006, Paulino *et al.* 2011) eles não fazem com que o experimento seja de fato passível de reprodução. Com os dados de proveniência, o cientista tem acesso, por exemplo, aos valores de parâmetros utilizados, porém, somente isso não é suficiente para reproduzir um *workflow*. Isso porque o ambiente no qual o *workflow* foi originalmente executado pode não existir mais, o mesmo vale para as versões dos programas e configurações dos mesmos. Uma das principais preocupações em termos de reprodução de experimentos é a questão da rápida evolução da infraestrutura dos sistemas de computação. Dada esta evolução, ainda será possível reexecutar o experimento concebido em uma tecnologia obsoleta?

Esse cenário ainda se torna mais complexo quando tratamos de uma reprodução de longo prazo. Tomemos por exemplo a execução de um *workflow* que precisa ser reproduzido daqui a 15 anos. As plataformas computacionais poderão ser outras, fazendo com que os programas hoje utilizados sejam incompatíveis com as futuras arquiteturas, além disso, muitos dos programas utilizados têm seus projetos descontinuados. Assim, preservar o ambiente original de execução ainda é um problema em aberto. Este é um problema ressaltado por Gil *et al.* (2007) e tratado em eventos tradicionais de banco de dados como o ACM SIGMOD (www.sigmod.org) que mantém uma trilha específica chamada *Repeatability section of the ACM SIGMOD* que tem como objetivo prover soluções que garantam a reprodução (ou repetição) dos experimentos submetidos a este congresso. Já existem algumas soluções que têm como objetivo final o compartilhamento de resultados de experimentos, mas, na maioria dos casos, não garantem a sua reprodução. Exemplos são o myExperiment (Goble *et al.* 2010) e o CrowdLabs (Mates *et al.* 2011), que possibilitam o compartilhamento de *workflows* científicos e anotações dos cientistas. Mas tanto o myExperiment quanto o CrowdLabs não permitem que o experimento seja reproduzido, apenas possibilitam o compartilhamento dos resultados, sendo responsabilidade dos cientistas montarem o ambiente (caso possível) para executar o *workflow*.

A computação em nuvem traz uma nova perspectiva para a reprodução de *workflows*. Ela é baseada na tecnologia de máquinas virtuais (VM) e fornece recursos teoricamente ilimitados (elasticidade). Uma VM pode ser definida como uma cópia isolada de um sistema físico, sendo ideal para possibilitar a reprodução de um experimento. A grande vantagem da virtualização é conseguir emular uma arquitetura e um sistema operacional em uma máquina física que nem sempre possui as mesmas características arquiteturais da máquina emulada. A abordagem apresentada nesse artigo é baseada na hipótese de que é possível reproduzir o ambiente de execução do experimento, e consequentemente dos resultados obtidos, utilizando a tecnologia de VM para atingir a reprodução de cada um dos ambientes envolvidos na execução do *workflow*. Portanto, o ambiente do experimento original é transformado em Imagens de Máquinas Virtuais (VMI) que podem ser disponibilizadas juntamente com a publicação de um artigo, fazendo com que os resultados sejam passíveis de reprodução por terceiros. Assim, uma VM é gerada de forma a manter as características de *software* originalmente definidos, e também é utilizada como base da infraestrutura de computação em nuvem. Um dos desafios se concentra em capturar as configurações dos mais diversos tipos de ambientes de forma a empacotá-las em uma VMI que possa ser instanciada e utilizada *a posteriori*, para a reprodução do experimento. Este artigo apresenta o ReproeScience (*i.e. Reproductible e-Science*), uma abordagem de reprodução de experimentos a ser agregada a um SGWfC, como o VisTrails (Callahan *et al.* 2006), ou a um SGWfC em nuvens, como o SciCumulus (Oliveira et al. 2010). A Seção 2 apresenta os trabalhos relacionados à reprodução/repetição e a nuvens de computadores. A Seção 3 descreve o ReproeScience, enfatizando os módulos de clonagem e reprodução do experimento. A Seção 4 destaca as conclusões parciais obtidas.

2. REPRODUÇÃO DE EXPERIMENTOS E NUVENS DE COMPUTADORES

Nos últimos anos, a computação em nuvem se tornou uma realidade no âmbito acadêmico. A idéia da computação em nuvem evoluiu do conceito de computação em grade aliada aos conceitos de escalabilidade e disponibilidade. Uma das vantagens da nuvem para os *workflows* é prover aos cientistas o acesso a uma variedade de recursos computacionais sem a necessidade de adquirir uma cara infraestrutura computacional. O ponto chave da nuvem é a sua divisão em camadas de abstração separadas por interfaces bem definidas através de VMs. Uma VM é definida como uma camada de *software* em um sistema de computação físico com o objetivo de se obter uma ou mais arquiteturas de VMs desejadas (Vaquero *et al.* 2009). Esta camada de *software*, denominada de sistema convidado, é executada sob o sistema operacional hospedeiro que possui um *hypervisor*, responsável por gerenciar esta camada de *software*. Utilizando as VMs oferecemos um ambiente independente e altamente configurável por cientista. São exemplos de *workflows* científicos executados em nuvem, análises filogenéticas (Ocaña et al. 2011) e de dados de astronomia (Hey *et al.* 2009). Consequentemente, é necessário assegurar a reprodução desses *workflows*. *Workflows* apoiam não somente a pesquisa em ciência da computação, mas também pesquisas em ciências naturais, ciências sociais e humanas. Por esse motivo, os *workflows* devem atender aos mesmos padrões de reprodutibilidade de experimentos de ciências naturais (Koop *et al.* 2011). Tal reprodução pode ser assistida por meio da captura de dados de proveniência. Dados de proveniência estão relacionados ao histórico do experimento, seja em relação à sua estrutura, sua execução ou ao ambiente no qual foi executado. Sendo este último tipo o objeto de estudo deste artigo. Algumas soluções já oferecem a captura de proveniência para validação dos experimentos como o PASS (Muniswamy-Reddy *et al.* 2009) e o SciCumulus (Oliveira et al. 2010) e poderiam ser utilizadas como base para uma abordagem que ofereça recursos de reprodução do experimento.

Esta necessidade de reprodução é um problema em aberto, porém algumas soluções já foram propostas nesse sentido, conforme apresentado a seguir. Na proposta de Macko *et al.* (2011) é apresentada uma extensão do PASS para coleta de proveniência de experimentos por meio do sistema operacional, utilizando um *hypervisor* Xen. Nessa abordagem, os autores coletam dados de proveniência por meio de sinais do sistema operacionais (SO), tais como interrupções e *system calls* e tem como objetivo recriar o ambiente utilizando estes dados de proveniência coletados. Entretanto, ao se capturar proveniência por meio do SO, informações como arquivos produzidos, configurações utilizadas, não podem ser capturadas. A abordagem Paper Mâché (Brammer *et al.* 2011) e a abordagem proposta por Koop *et al.* (2011), ambos apresentados no Grande Desafio de Artigos Executáveis da Elsevier (Gabriel e Capone 2011), propõem soluções para artigos com reprodução de experimentos garantida. Enquanto que o Paper Mâché utiliza VMs, o que permite que os leitores e avaliadores facilmente visualizem e interajam com um artigo, a abordagem de Koop *et al.* foca em associar um *workflow* no VisTrails a cada artigo. Estas abordagens viabilizam a reprodução do experimento, mas o cientista deve configurar todo o ambiente, pois tais abordagens apenas tem como objetivo disponibilizar as informações de reprodução do *workflow* como parâmetros e dados. Esta configuração manual do experimento pode ser complicada e suscetível a erros. Além disso, por meio destas abordagens, não podemos executar *workflows* em paralelo, somente em modo sequencial.

3. REPROESCIENCE

Para a reprodução efetiva de experimentos científicos, modelados como *workflows* e executados pelos SGWfC, se torna fundamental possuir o mesmo ambiente computacional da execução original. Assim, é necessário reconstruir o ambiente a partir dos dados de proveniência e, conforme proposto neste artigo, por meio de VMI. Para tal, é proposto o *middleware* ReproeScience. A Figura 1 apresenta a arquitetura do ReproeScience, que é utilizada por meio de uma interface que possui três métodos principais, responsáveis por obter os dados de proveniência, clonar o ambiente no qual o experimento

foi executado e reconstruir os sistemas de computação envolvidos na execução do *workflow*. O ReproeScience é invocado pelos SGWfC através de funções definidas pelo usuário (do inglês *User Defined Functions* ou UDF). O ReproeScience fornece apoio para as UDF como uma forma de especificar o processamento personalizado. Além disso, o ReproeScience armazena os dados necessários para autenticação com provedores de nuvem. A execução do ReproeScience inicia-se juntamente com a execução do *workflow* científico. À medida que o *workflow* é executado, dados de proveniência são gerados e os mesmos são utilizados para a recuperação das configurações do ambiente pelo ReproeScience. Após o término da execução do *workflow*, a abordagem solicita ao cientista a opção de clonagem do ambiente, e uma vez que ele opte por iniciar este processo (muitas vezes demorado), as VMI são geradas, tornando assim possível a reprodução do experimento, sob iguais condições, ao executado originalmente.

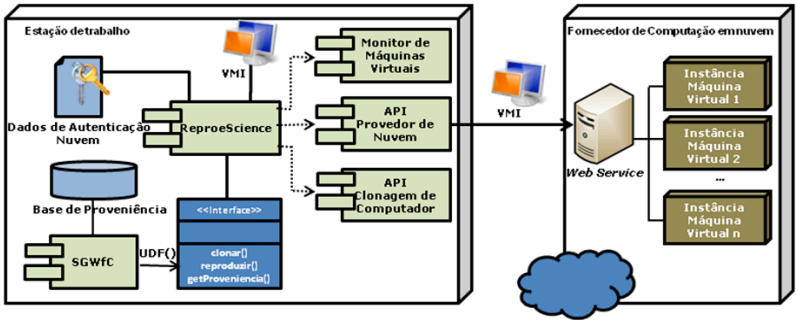

Figura 1 Arquitetura do ReproeScience

O módulo de clonagem do sistema é utilizado para capturar as informações do ambiente original e gerar as VMIs. Para clonar o ambiente é necessário utilizar uma ferramenta de clonagem, tal como o Clonezilla (http://clonezilla.org/). Além disso, é necessário ter um Monitor de Máquinas Virtuais (VMM) para criar as VMIs e ambos componentes devem estar instalados na máquina que executa o SGWfC. Atualmente, o ReproeScience trabalha com VMs nos padrão do VMM Xen. O diagrama de atividades da Figura 2 apresenta as atividades executadas pelo módulo de clonagem. A coleta de proveniência é realizada pela máquina de execução de *workflows* a qual o ReproeScience está acoplado, e é obtida por meio da interface de comunicação. Cabe a máquina de execução escolhida obter os metadados do ambiente para a geração das imagens das máquinas virtuais. Ao término da execução do *workflow*, a atividade de finalização obtém os dados de proveniência do ambiente do experimento e define um objeto chamado "experimento" no ReproeScience. Neste momento são finalizados os registros de proveniência e demais características da execução do *workflow*. Após a atividade de finalização fica sob a responsabilidade de o cientista informar se deseja ou não criar as VMI dos sistemas de computação envolvidos na execução do *workflow*. É importante frisar que um dado *workflow* pode executar em mais de um ambiente, como um *desktop* e um cluster e o VMM deve registrar os dados de todos estes ambientes para a criação da VMI. Caso o cientista não deseje registrar a VMI para a reprodução, o objeto "experimento" é destruído e o ReproeScience é finalizado.

Caso o cientista opte pela geração das VMs, o ReproeSience executa uma atividade de clonagem do ambiente original, utilizando os dados de proveniência armazenados no objeto "experimento", por meio do VMM para gerar as VMI (por este motivo é preciso ter o VMM instalado). Após a solicitação da clonagem, o ReproeScience fica aguardado a resposta do VMM. Caso o processo de geração das imagens falhe, uma atividade de informação de erro é executada e o processo de clonagem é finalizado. Caso a geração das imagens seja bem sucedida um objeto chamado "virtualMachineImage" é criado e o mesmo é agregado ao "experimento". Este objeto ao final da execução do *workflow* contém todos os dados de proveniência necessários para a clonagem do ambiente computacional no qual o *workflow* foi executado. É importante ressaltar que o ReproeScience objetiva clonar ambientes com muitos processadores, como um cluster, por exemplo. Desta forma, ele garante que as imagens que são produzidas sejam consistentes entre as diversas maquinas que compõem o ambiente.

No caso das informações sobre o *hardware*, os dados são coletados pelo VMM, tais como: as características de unidade de processamento (CPU), a descrição da memória e disco rígido (capacidade), a arquitetura do sistema (centralizada, cliente-servidor, cluster e *etc.*), e informações sobre a rede de computadores. Apesar de uma VMI não conter informações referentes ao *hardware* (isso somente é definido no momento da instanciação das máquinas virtuais) o objeto "virtualMachineImage" contém os metadados associados ao *hardware* ideal para que a máquina virtual a ser criada *a posteriori* seja equivalente ao ambiente originalmente executado. Em termos de *software*, é necessário obter informações sobre o sistema operacional, com sua base operacional (ex. 32 ou 64 bits), os programas e bibliotecas invocadas durante a execução do *workflow*. Deve-se atentar ao uso de serviços *Web*, pois determinadas atividades do *workflow* científico podem executar processamento em tais serviços.

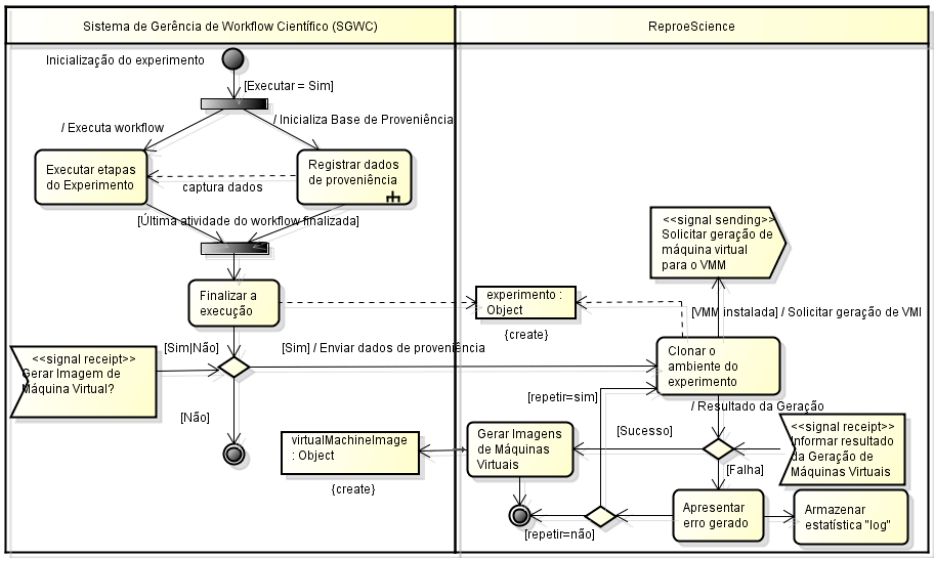

Figura 2 Atividades do Módulo de Clonagem

Uma vez que as imagens de máquinas virtuais estão prontas, o processo de reconstrução pode ser executado através do módulo de reconstrução. O ambiente original pode ser reconstruído de duas formas diferentes: (i) reproduzido em uma estação de trabalho ou (ii) reproduzido na nuvem. Atividades computacionalmente menos custosas podem ser executadas localmente enquanto as mais custosas podem ser executadas na nuvem. Com a utilização de VMs, o ambiente pode ser instanciado em qualquer uma das arquiteturas de forma transparente. Logicamente, se o cientista necessitar de alto poder de processamento a nuvem se torna a única opção, uma vez que oferece recursos de forma elástica. No contexto de computação em nuvem é necessário possuir acesso por meio do sistema de autenticação do provedor de nuvem, por exemplo possuir um par de chaves para autenticação que são expressos no SLA (*Service Level Agreement*). Tais informações estão presentes na base de proveniência e são disponibilizadas para o cientista no momento da reconstrução da máquina virtual.

Outra característica que é explorada no desenvolvimento é a capacidade de incorporar múltiplos provedores de nuvem na reprodução do *workflow*. O objetivo é utilizar instâncias de VMs em diferentes provedores de nuvem para a execução do *workflow* científico. Tal característica reforça o conceito de "recursos ilimitados", pois se a demanda de recursos ultrapassar a capacidade de recursos de um provedor, o *middleware* pode passar parte da carga para outros provedores de nuvem. Na Figura 3, podemos verificar que o ReproeScience inicia-se juntamente com o SGWfC. Para solicitar a reprodução do experimento, o cientista seleciona o *workflow* que deseja reproduzir e em seguida os dados de proveniência de reprodução são recuperados. Existe uma diferença entre os dados de

proveniência do *workflow* original e dos ambientes clonados. Para o módulo de reconstrução, os dados importantes são os relacionados às VMs e VMI criadas (proveniência de reprodução). Baseado nestes dados são selecionadas a(s) VMI que podem ser instanciadas para esta execução. As VMI que possibilita a execução do *workflow* são carregadas e instanciadas. Após o carregamento das VMI existe um ponto de decisão onde o cientista pode selecionar em que ambiente o *workflow* será reproduzido (localmente ou na nuvem).

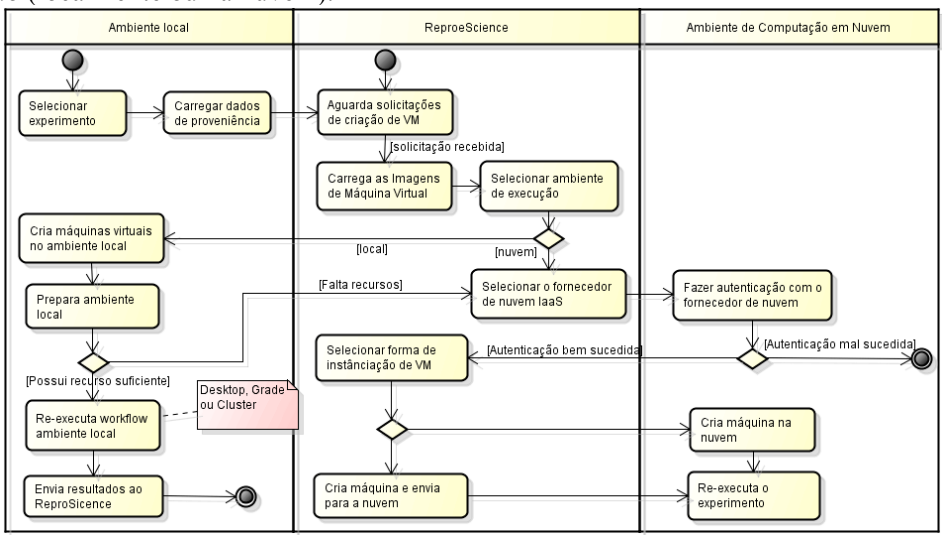

Figura 3 Atividades do Módulo de Reconstrução

Para garantir que as informações de proveniência coletadas nos diferentes SGWfC pudessem ser armazenadas e utilizadas pelo ReproeScience foi desenvolvido um modelo de dados baseado no *Open Provenance Model* (OPM) (Moreau *et al.* 2008). Todavia, foram observadas características para adequar o Modelo de dados do ReproeScience ao PROV do grupo de trabalho de proveniência do W3C, uma vez que o PROV é uma evolução do OPM. Os protocolos de adequação ao padrão são definidos e executados pelo objeto "experimento". O módulo de proveniência do ReproeScience executa atividades de captura de dados da base de proveniência dos SGWfC e ainda obtém dados necessários para a reprodução com VMs. A Figura 4 apresenta o modelo de proveniência do ReproeScience representado como um diagrama de classes da UML. Este modelo contém informações tanto do *workflow* quanto do ambiente clonado. Inicialmente, pode se verificar que a classe "***Workflow***" é parte integrante de um "**Experimento**" e possui dados como data de geração, nome e versão, todos obtidos da base de proveniência do SGWfC. As características da classe "Autor" também são obtidas através de informações do SGWfC. Em seguida, pode-se observar as classes "**AutenticacaoNuvem**" e "**FornecedorNuvem**", as quais possuem os dados para executar a gerência do ambiente de reprodução em um ou vários fornecedores de nuvem. Nestas classes são identificadas as chaves de acesso e as informações (porta e IP) para a conexão com o provedor de nuvem.

Seguindo o modelo da Figura 4, pode-se observar que um *workflow* é composto por uma lista de **"Atividades"** que representam os processos executados nas etapas do *workflow*. Pode-se verificar que as atividades possuem objetos de pesquisa – artefatos – ligados a si, os quais são representados pela classe "**ObjetoPesquisa**". Esta classe é uma generalização que se desmembra em diversos outros objetos especializados, tais como os arquivos de entrada de dados, os resultados intermediários entre as etapas do *workflow*, os programas e bibliotecas invocados em tempo de execução, enfim, todos os elementos que compõem um experimento modelado como um *workflow*. Porém, o diagrama foca os dados para a reconstrução das VM, que são representadas pelo objeto "**ImagemMaquinaVirtual**". Neste objeto é identificado o *hypervisor*, ou seja, o VMM, os dados de identificação, tais como o nome e o caminho contido no PATH, a senha de *root* e o endereço IP, este último obtido do SGWfC e

necessário para se verificar a necessidade de clonar uma determinada quantidade de diferentes ambientes.

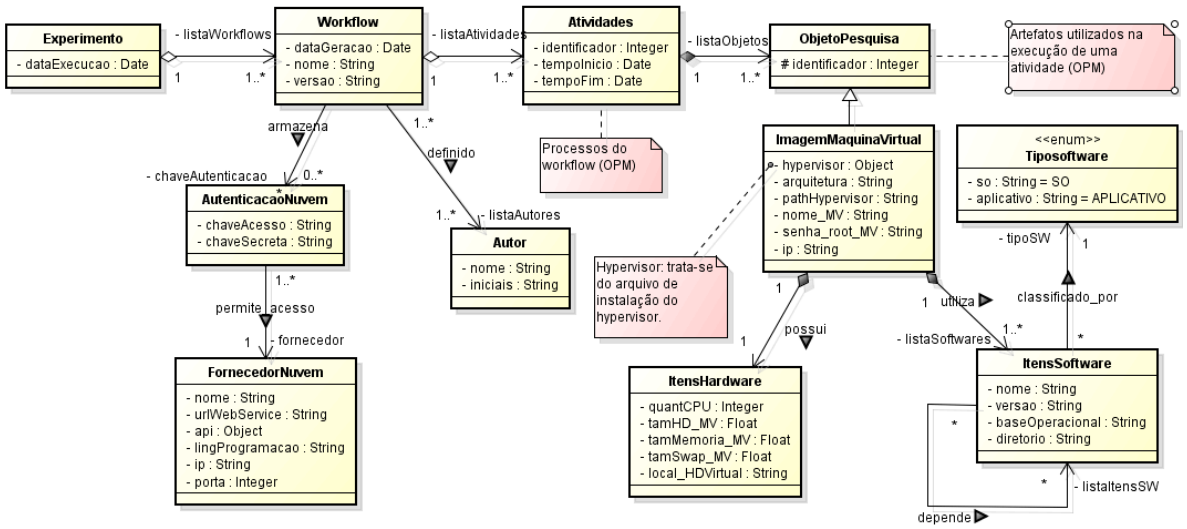

Figura 4 Classes do Modelo de Proveniência do ReproeScience

Vale destacar que o cientista deve ter privilégios de administrador para conseguir clonar os ambientes. Ainda na Figura 4, o objeto "ImagemMaquinaVirtual" possui as classes "**ItensHardware**" e "**ItensSoftware**" como classes integrantes, as quais identificam as características do ambiente operacional de execução do *workflow*. No caso de "**ItensHardware**" são identificadas o tamanho da memória, HD, CPU e *Swap* da arquitetura do sistema de computação. Já no caso do objeto "**ItensSoftware**" é identificada uma lista de *softwares* invocados para o processamento do *workflow*, bem como suas características e a dependência existente entre os *softwares*, identificada através de chamadas em tempo de execução.

4. CONCLUSÕES

Nos últimos anos a comunidade tem empreendido uma série de esforços para alcançar a reprodução de experimentos em diversas comunidades científicas. Todo esse esforço provavelmente levará à criação de grandes repositórios de experimentos reproduzíveis. Entretanto, reproduzir um experimento científico em sua plenitude ainda é um desafio em aberto, especialmente no que se refere às características do ambiente em que o experimento foi originalmente executado. Neste artigo apresentamos o ReproeScience, uma abordagem que visa à reprodução de um experimento científico modelado como um *workflow* por meio de máquinas virtuais e computação em nuvem como arcabouço básico para a reprodução do ambiente original em que o experimento foi executado. A utilização da nuvem computacional traz como vantagem o acesso a recursos elásticos e com alta disponibilidade. O ReproeScience está em fase de desenvolvimento e existe um protótipo implementado na linguagem Java que utiliza o sistema operacional Ubuntu 10.10 para realizar a clonagem e a geração da VM utilizando o *hypervisor Xen*. Atualmente, o código da clonagem do ambiente está sendo depurado e estabilizado. O provedor de nuvem utilizado neste protótipo é o da Amazon EC2, todavia, estamos executando testes com o GoGrid de forma a estender o número de provedores de nuvem e até mesmo incluir processamento de uma atividade com dois provedores cooperando um com o outro.

Apesar de ser um trabalho em andamento, essa primeira proposta abre um vasto campo de pesquisas futuras, porém testes ainda devem ser realizados com a abordagem proposta. Para realização de testes dessa natureza, está planejada a execução do *workflow* SciPhy (Ocaña *et al.* 2011) como estudo de

caso na máquina de *workflows* SciCumulus. O SciPhy tem como objetivo analisar a relação evolutiva entre uma série de organismos que são informados pelos cientistas. Este *workflow* necessita de um ambiente de processamento de alto desempenho, uma vez que é composto de 1.250 atividades paralelas que demandam processamento paralelo. O objetivo final do ReproeScience é prover toda a infraestrutura necessária para a reprodução dos experimentos a longo prazo. Por meio da utilização desta infraestrutura, estaremos aptos a criar um repositório de experimentos reproduzíveis que possam ser instanciados pelos cientistas de forma simples, sem necessidade de realizar complexas configurações de ambiente. As técnicas de banco de dados nos fornecem subsídios para a modelagem dos dados de proveniência, criação dos repositórios de experimentos e para a exploração das informações que eles detêm.

5. REFERÊNCIAS

Altintas, I., Barney, O., Jaeger-Frank, E., (2006), "Provenance Collection Support in the Kepler Scientific Workflow System", *Provenance and Annotation of Data*, , chapter 4145, Springer Berlin, p. 118-132.

Brammer, G. R., Crosby, R. W., Matthews, S., Williams, T. L., (2011), "Paper Mâché: Creating Dynamic Reproducible Science.", *Procedia CS*, v. 4, p. 658-667.

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., Vo, H. T., (2006), "VisTrails: visualization meets data management". In: *SIGMOD International Conference on Management of Data*, p. 745-747, Chicago, Illinois, 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.

Gabriel, A., Capone, R., (2011), "Executable Paper Grand Challenge Workshop", *Procedia Computer Science*, v. 4, p. 577-578.

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., et al., (2007), "Examining the Challenges of Scientific Workflows", *Computer*, v. 40, n. 12, p. 24-32.

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M., et al., (2010), "myExperiment: a repository and social network for the sharing of bioinformatics workflows", *Nucleic Acids Research*, v. 38, n. Web Server Issue (jul.), p. 677-682.

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

Jarrard, R. D., (2001), *Scientific Methods*. Online book, Url.: http://emotionalcompetency.com/sci/booktoc.html.

Koop, D., Santos, E., Mates, P., Vo, H. T., Bonnet, P., Bauer, B., Surer, B., Troyer, M., Williams, D. N., et al., (2011), "A Provenance-Based Infrastructure to Support the Life Cycle of Executable Papers", *Procedia Computer Science*, v. 4, p. 648-657.

Macko, P., Chiarini, M., Seltzer, M., (2011), "Collecting Provenance via the Xen Hypervisor". In: *Proc. of TaPP'11TaPP'11*, Boston, United States.

Mates, P., Santos, E., Freire, J., Silva, C. T., (2011), "CrowdLabs: Social Analysis and Visualization for the Sciences". In: *23rd Scientific and Statistical Database Management Conference23rd Scientific and Statistical Database Management Conference*, Portland, Oregon, USA.

Mattoso, M., Werner, C., Travassos, G. H., Braganholo, V., Murta, L., Ogasawara, E., Oliveira, D., Cruz, S. M. S. da, Martinho, W., (2010), "Towards Supporting the Life Cycle of Large-scale Scientific Experiments", *International Journal of Business Process Integration and Management*, v. 5, n. 1, p. 79–92.

Moreau, L., Freire, J., Futrelle, J., McGrath, R., Myers, J., Paulson, P., (2008), "The Open Provenance Model: An Overview", *Provenance and Annotation of Data and Processes*, , p. 323-326.

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

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", In: Norberto de Souza, O., Telles, G. P., Palakal, M. [orgs.] (eds), *Advances in Bioinformatics and Computational Biology*, , chapter 6832, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 66-70.

Oliveira, D., Ogasawara, E., Baião, F., Mattoso, M., (2010), "SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows". In: *3rd International Conference on Cloud Computing*, p. 378–385, Washington, DC, USA.

Paulino, C. E., Cruz, S. M. S., Oliveira, D., Campos, M. L. M., Mattoso, M., (2011), "Capturing Distributed Provenance Metadata from Cloud-Based Scientific Workflows", *Journal of Information and Data Management*, v. 2, n. 1, p. 43-50.

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