UMA ABORDAGEM PARA ALTERAÇÃO NA DEFINIÇÃO DE ATIVIDADES DE WORKFLOWS CIENTÍFICOS EM TEMPO DE EXECUÇÃO

Igor de Araujo dos Santos

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadora: Marta Lima de Queirós Mattoso

Rio de Janeiro

Março de 2014

UMA ABORDAGEM PARA ALTERAÇÃO NA DEFINIÇÃO DE ATIVIDADES DE WORKFLOWS CIENTÍFICOS EM TEMPO DE EXECUÇÃO

Igor de Araujo dos Santos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____

Prof[a]. Marta Lima de Queirós Mattoso, D.Sc.

_____

Prof. Alexandre de Assis Bento Lima, D.Sc.

_____

Prof. Daniel Cardoso Moraes de Oliveira, D.Sc.

_____

Prof. Leonardo Guerreiro Azevedo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2014

Santos, Igor de Araujo dos

Uma abordagem para alteração na definição de atividades de workflows científicos em tempo de execução / Igor de Araujo dos Santos. – Rio de Janeiro: UFRJ/COPPE, 2014.

VIII, 45 p.: il.; 29,7 cm.

Orientadora : Marta Lima de Queirós Mattoso

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2014.

Referências Bibliográficas: p. 41-45.

1. *Workflows* Científicos Dinâmicos. 2. Interatividade. 3. Iteração. I. Mattoso, Marta Lima de Queirós II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Muitas vezes a caminhada é difícil e cheia de obstáculos. Por vezes pensamos que não somos capazes de chegar até o final. Mas o apoio, carinho e orientação de muitas pessoas fizeram com que eu acreditasse que poderíamos juntos alcançar um bom resultado. Quero, além de dizer um sincero "Muito Obrigado", ter condições de agradecer à altura tudo o que me foi proporcionado por estas pessoas ao longo de toda esta caminhada.

Primeiramente agradeço a Deus pela vida e por ter sido fonte de fé e inspiração em todos os momentos, principalmente nos momentos mais difíceis. Agradeço a Ele por ter colocado tantas pessoas boas no meu caminho.

Agradeço aos meus pais, José e Marlina, por toda a orientação que me deram para a vida. Por terem sido incansáveis ao me dar todo o apoio para que chegássemos ao fim deste trabalho. Agradeço ao meu irmão, Davi, por ter sido sempre meu companheiro, me apoiando nas minhas decisões e por, mesmo sendo irmão mais novo, ser um exemplo para mim. Agradeço à Aline, pelo grande apoio e incentivo na fase de conclusão deste trabalho. Agradeço a todos os meus familiares, pela união, pela preocupação e pelo apoio que me deram.

Agradeço à minha orientadora, Professora Marta Mattoso, pela oportunidade de desenvolver este trabalho, pela dedicação, pela atenção e pela confiança. Agradeço pela sua orientação, pelas sugestões, pela tolerância, pelas discussões e pelas ideias que tanto agregaram a este trabalho.

Um agradecimento especial ao Jonas Dias, Daniel Oliveira e Kary Ocaña pela amizade e por toda a co-orientação e ajuda que me deram ao longo de todo o trabalho. Muito obrigado pelos comentários, dicas, revisões, ajustes em textos, co-autoria em artigos e por serem também exemplos de pesquisadores para mim. Agradeço a todo o grupo de trabalho de e-science da COPPE, pelas trocas de ideias, pelas sugestões e pelo apoio nos artigos e nas apresentações em congressos.

Aos amigos da GPE, agradeço pela compreensão e incentivo durante todo o tempo em que trabalhamos juntos. Obrigado pela compreensão nas trocas de horário, pelas discussões sobre temas para pesquisa, por permitir utilizar o ambiente da empresa para realização de algumas tarefas deste trabalho. Um agradecimento especial à Paula Nascimento, por sempre ter sido uma grande incentivadora deste trabalho e por ter sido grande amiga e companheira em todo esse tempo.

A todos os meus amigos, agradeço pela compreensão nos fins de semana que não pude sair, por aceitar que muitas vezes não pude fazer parte de novos projetos por conta da dedicação que este trabalho requeria e por, acima de tudo, estarem ao meu lado e acreditarem em mim até mesmo quando eu desacreditava.

Aos funcionários da COPPE, muito obrigado pelo bom trabalho e por estarem diariamente trabalhando para que nós, alunos, possamos desenvolver nossas pesquisas. A toda a equipe administrativa do PESC, muito obrigado por todo o suporte e pelo bom trabalho.

Agradeço ao CNPq pela concessão da bolsa de mestrado entre março de 2011 e fevereiro de 2013.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA ABORDAGEM PARA ALTERAÇÃO NA DEFINIÇÃO DE ATIVIDADES DE *WORKFLOWS* CIENTÍFICOS EM TEMPO DE EXECUÇÃO

Igor de Araujo dos Santos

Março/2014

Orientadora: Marta Lima de Queirós Mattoso

Programa: Engenharia de Sistemas e Computação

Os *workflows* científicos fornecem uma abstração para a representação de experimentos baseados em simulações computacionais. Em geral, *workflows* executados em paralelo manipulam uma grande massa de dados e demandam um elevado tempo de execução, o que pode dificultar ou encarecer o processo exploratório de um experimento. Em um *workflow* científico podem existir atividades que podem ser executadas por diferentes programas, aos quais chamamos de variabilidades. De acordo com os requisitos do experimento e com o ambiente de execução, que podem ser alterados durante a execução de um *workflow*, a utilização de uma variabilidade pode ser mais vantajosa do que outra. Diante deste cenário, tornou-se desejável a utilização de *workflows* com aspectos dinâmicos, que permitem que mudanças na definição do *workflow* possam ser realizadas em tempo de execução a fim de atender melhor aos requisitos do experimento a qualquer momento da execução. Assim, o objetivo desta dissertação é oferecer uma abordagem para permitir alterações na definição de atividades de *workflows* em tempo de execução, por meio da troca dinâmica de uma atividade pela variabilidade que mais bem atenda aos requisitos do experimento, permitindo assim a interferência do cientista, a condução dinâmica da execução do *workflow* e, como consequência, reduzir o tempo e o custo monetário associados à execução do *workflow*.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTERACTIVE EXECUTION FOR LARGE SCALE COMPUTATIONAL SCIENTIFIC EXPERIMENTS

Igor de Araujo dos Santos

March/2014

Advisor: Marta Lima de Queirós Mattoso

Department: Systems and Computer Engineering

Scientific workflows provide an abstraction to represent experiments based on computer simulations. Generally, parallel workflows handle a huge amount of data and require a long time to be executed, which can hamper or endear the exploratory process of an experiment. In a scientific workflow some activities can be implemented by different programs, that we call variability. According to the requirements of the experiment and the execution environment, that may change during the execution of a workflow, the use of a variability may be more advantageous than another. In this scenario, has become desirable the use of workflows with dynamic features that allow changes in the workflow definition that can be performed at runtime in order to better meet the requirements of the experiment at any time of execution. The goal of this dissertation is to provide an approach to allow changes in the definition of workflow activities at runtime, through dynamic exchange of an activity by the variability that best meets the requirements of the experiment, thus allowing the interference of the scientist, the dynamic user steering of the workflow execution and as a consequence, reduce the time and financial cost associated with the execution of the workflow.

Sumário

1. Introdução

A experimentação científica é um artifício utilizado pelo homem na investigação de fenômenos, com o objetivo de adquirir novos conhecimentos ou corrigir e integrar conhecimentos previamente estabelecidos (Wilson Jr 1991). A experimentação científica é uma das formas usadas para apoiar as teorias baseadas em um método científico (Jarrard 2001). Um experimento consiste na elaboração de um protocolo concreto a partir do qual se organizam diversas ações observáveis, direta ou indiretamente, de forma a corroborar ou refutar uma dada hipótese científica a qual foca em estabelecer relações de causa/efeito entre fenômenos observáveis (Beveridge 2004, Jr 2002).

Com o passar do tempo e com o aumento da complexidade dos experimentos, estes passaram a ser executados em larga escala, demandando uma maior capacidade de gerenciamento. Os avanços na ciência da computação, como o aumento da capacidade de processamento, a evolução na infraestrutura de rede e novas tecnologias de armazenamento, permitiram o desenvolvimento de experimentos científicos complexos baseados em simulação, que envolvem a manipulação e análise de uma grande massa de dados, dando origem a uma categoria de experimentos chamada *in silico* (Travassos e Barros 2003). A partir dessa categoria de experimentos surge o conceito de *e-science* (ou e-Ciência), que é o uso intensivo de computação para modelagem e execução de experimentos científicos, sejam eles de larga escala ou não. A *e-science* oferece ao cientista um apoio ao desenvolvimento em larga escala por meio de uma infraestrutura computacional correspondente (Mattoso *et al.* 2008).

Os experimentos *in silico* são caracterizados pelo uso de simulações de eventos reais da natureza e são constituídos tipicamente de modelos computacionais, que são explorados a fim de se conhecer as características do objeto que está sendo estudado (Mattoso *et al.* 2010). Em geral estes modelos envolvem um encadeamento de programas que são invocados e executados durante as simulações, onde cada execução pode consumir e produzir uma grande massa de dados. Muitos desses programas fazem uso intenso de recursos computacionais e sua execução pode se tornar inviável caso o volume de dados a ser processado seja demasiadamente grande. Desta forma, para que o experimento seja executado, é necessária uma infraestrutura de processamento paralelo (Culler *et al.* 1999). O encadeamento desses programas para uma simulação não é uma

tarefa trivial e em alguns casos pode inviabilizar a execução do experimento (Gil *et al.* 2007). Gerenciar experimentos científicos em larga escala, garantindo sua reprodutibilidade e consistência é uma tarefa importante e complexa (Freire *et al.* 2008).

A execução de experimentos baseados em simulação pode ser apoiada por *workflows* científicos (Deelman et al. 2009, Taylor et al. 2007a). Os *Workflows* científicos são usados como uma abstração para representar o processo de um experimento científico computacional, definindo o encadeamento de programas necessários para executar o experimento. Um *workflow* científico pode ser definido como um grafo, onde os vértices são as atividades, que representam os programas utilizados no experimento, e arestas, que são as dependências de dados entre as atividades.

Ao longo do tempo, com a crescente complexidade dos experimentos, grades computacionais, *clusters* e, mais recentemente, nuvem de computadores, passaram a ser utilizados como ambientes de execução de *workflows* científicos, lançando novos desafios relacionados à gerência da execução dos experimentos em ambientes distribuídos (Gil *et al.* 2007). Para a composição, execução e análise dos *workflows* científicos em larga escala, cientistas utilizam ferramentas como os Sistemas de Gerência de *Workflows* Científicos (SGWfC) com capacidades de processamento paralelo, como o Swift/Turbine (Wozniak *et al.* 2012, Zhao *et al.* 2007) o Pegasus (Deelman *et al.* 2007), o SciCumulus (Oliveira *et al.* 2010), o Chiron (Ogasawara *et al.* 2013), o Triana (Taylor *et al.* 2007b) e o Tavaxy (Abouelhoda *et al.* 2012).

Os SGWfCs (Altintas *et al.* 2004, Callahan *et al.* 2006) são ferramentas que permitem a composição, execução e análise dos experimentos científicos. Um aspecto importante de um SGWfC é a gerência dos dados de proveniência (Freire *et al.* 2008). A proveniência permite uma análise mais precisa dos resultados e permite a reprodução do experimento por outros cientistas favorecendo o processo de expansão da ciência. Os SGWfCs existentes possuem recursos para dar apoio ao processamento paralelo e de alto desempenho e a gerência dos dados de proveniência. Entretanto, conforme enfatizado por Gil *et al.* (2007) , ainda existem problemas em aberto que podem dificultar o uso efetivo dos SGWfCs em domínios importantes da ciência em larga escala, principalmente no que diz respeito ao apoio a *workflows* dinâmicos. Um *workflow* dinâmico é um conceito projetado para dar apoio à natureza exploratória da ciência e ao processo dinâmico envolvido nas análises científicas (Gil *et al.* 2007). De

acordo com Gil *et al.* (2007), o apoio a *workflows* dinâmicos inclui recursos como adaptação do *workflow* baseada em eventos externos como a intervenção humana e condução dinâmica. Um recurso necessário é permitir que o cientista realize alterações dinâmicas na especificação do *workflow* quando este já está em execução em um ambiente remoto.

Em experimentos científicos, até que se alcance o resultado desejado, os *workflows* podem passar por uma série de alterações na sua especificação e nos seus parâmetros para que se atinja a configuração que produza o resultado desejado para o experimento (Dias *et al.* 2011). Muitas destas alterações são realizadas mediante a análise dos resultados produzidos ao longo da execução do *workflow*. Um cenário muito comum é aquele em que o cientista analisa o resultado de uma primeira execução do *workflow* e então toma decisões sobre o que será feito em seguida. Ou seja, de acordo com resultados de execuções anteriores, o cientista decide o que deverá ser ajustado no *workflow* para a próxima execução. Diante disso, um recurso interessante seria o acesso à proveniência em tempo de execução, o que permitiria ao cientista fazer a análise dos resultados parciais de um *workflow*, possibilitando a tomada de decisões sobre que alterações poderiam ser necessárias antes mesmo do término da execução do workflow. A proveniência em tempo de execução fornece a base para que as alterações no workflow também possam ser realizadas em execução. Uma opção de alteração é modificar os parâmetros e dados de entrada até que o objetivo da sua execução seja alcançado. Os dados de entrada e os parâmetros têm influência direta nos resultados gerados, o que leva os cientistas a explorarem diferentes conjuntos de dados e parâmetros e executarem o *workflow* repetidas vezes para descobrir qual a melhor configuração que produzirá o resultado desejado. De acordo com experimentos anteriores (Dias *et al.* 2011), evidenciamos o potencial de características dinâmicas em *workflows* para monitoramento, avaliação e adaptação da execução de um *workflow* com necessidade de ajustes em seus parâmetros.

Entretanto, não são apenas os parâmetros e os dados de entrada que precisam de ajustes. Muitas vezes os cientistas, baseados nos resultados de execuções anteriores, realizam modificações na especificação do *workflow*, alterando suas atividades e relacionamentos a fim de obter um resultado diferenciado ou mais adequado. No contexto de experimentos executados por meio de *workflows*, podem ocorrer casos em

que uma mesma atividade do *workflow* tenha várias implementações (*i.e.* programas) correspondentes.

Para exemplificar essa necessidade, tomemos como exemplo o experimento de análise filogenética. Esse exemplo será usado consistentemente ao longo dessa dissertação. A análise filogenética (Eisen 2003) é uma das muitas áreas da bioinformática que tem como foco a comparação de centenas de diferentes genes a fim de identificar similaridade dentre os diferentes organismos. A gerência de experimentos de análise filogenética não é trivial, uma vez que suas atividades são computacionalmente intensivas e produzem uma grande massa de dados. Especialmente em *workflows* de análise filogenética, cientistas executam um conjunto específico de atividades para produzir um conjunto de árvores filogenéticas que são usadas para inferir a relação de evolução entre genes de diferentes espécies. O SciPhy (Ocaña *et al.* 2011b), *workflow* que será usado como estudo de caso dessa dissertação, é um exemplo de *workflow* de análise filogenética. Este *workflow* usa varredura de parâmetros onde todas as atividades são executadas para cada arquivo em um dado conjunto de dados de entrada. O SciPhy é composto por quatro atividades. Para cada uma dessas atividades, existem múltiplos programas que a implementam. Uma execução típica do SciPhy pode levar dias para chegar ao seu fim, e se for executado em ambientes de computação de alto desempenho, como nuvens de computadores (Vaquero *et al.* 2009), o custo financeiro é também uma questão importante já que os provedores de nuvem cobram por unidade de tempo. Em cada experimento, o cientista pode ter restrições como o tempo máximo esperado para execução, o custo financeiro máximo, ou algum critério de qualidade sobre os resultados gerados. Atualmente, a execução em ambientes distribuídos como a nuvem, tem foco na otimização do escalonamento das atividades ou políticas de distribuição dos dados. Entretanto, baseado na condução do usuário e na análise parcial dos dados, alterações dinâmicas dos programas associados às atividades podem ser realizadas com a finalidade de melhorar o desempenho e aumentar a qualidade dos resultados gerados. Por exemplo, uma atividade de Alinhamento Múltiplo de Sequências (AMS) pode utilizar diferentes programas. Assim, estes programas podem ser vistos como alternativos entre si. De acordo com o ambiente de execução e com os requisitos do experimento, um programa pode ser mais vantajoso do que outro devido às características da sua implementação. Levando-se em conta a execução na nuvem, um *workflow* pode ter desempenho e resultados variados de acordo com o

número de máquinas virtuais instanciadas, características das imagens utilizadas, etc. Ou seja, existem aspectos que podem influenciar a execução do *workflow*. Cada alternativa para uma determinada atividade pode ter um comportamento diferente de acordo com estes aspectos. Além disso, vale lembrar que estes aspectos podem variar durante a execução do *workflow*. Por exemplo, a capacidade de processamento (número de processadores disponíveis) de um ambiente onde o *workflow* está sendo executado pode variar durante a execução.

Baseado neste cenário surge a necessidade de um apoio dinâmico para a escolha de uma atividade alternativa, pois a melhor configuração não pode ser conhecida *a priori*, uma vez que os requisitos e o ambiente podem sofrer mudanças. Deve ser possível alterar as atividades de um *workflow* em execução sem que se faça necessário parar e reexecutar o *workflow* por completo para que as alterações sejam consideradas. Por exemplo, consideremos que o SciPhy teve sua execução iniciada com a escolha do programa MAFFT para a atividade de Alinhamento Múltiplo de Sequência(AMS). Após um determinado tempo de execução, o cientista recebe uma notificação da nuvem informando-o de que o tempo gasto na execução desta atividade está sendo maior que o esperado. Alguma ação deve ser tomada para atender à restrição do cientista. Seria interessante se a máquina de *workflow* pudesse mudar o programa utilizado automaticamente e sem necessidade de reexecução, baseado na sugestão do cientista para alcançar um melhor custo benefício na execução do alinhamento de sequências.

De acordo com as abordagens atuais, os cientistas precisam interromper ou esperar o término da execução para realizar ajustes no *workflow* para então reexecutá-lo por completo. Assim, o cientista pode ter que esperar por um longo período até poder realizar as análises, que também tendem a ser mais demoradas, devido à grande massa de dados, e só então ajustar o *workflow* para executá-lo novamente. Atrelado ao tempo de execução pode existir o custo financeiro, pois o tempo utilizado para execução de um *workflow* em uma nuvem computacional, por exemplo, é cobrado sob demanda. Neste cenário, um tratamento dinâmico para os *workflows* científicos permite que os cientistas realizem ajustes no *workflow* sem que tenham que esperar o término de uma execução para efetuar alguma alteração e, só então, executar novamente. Um dos objetivos de trabalhar com *workflows* dinâmicos é permitir que os ajustes realizados sejam refletidos em tempo real sobre a(s) instância(s) em execução de um *workflow*, com o intuito de apoiar o cientista a obter os resultados finais do experimento mais rapidamente.

A abordagem proposta nesta dissertação, DynAdapt, está direcionada ao apoio às mudanças na especificação de *workflows* científicos. O DynAdapt possui um conjunto de operações para dar apoio às máquinas de *workflow* na modificação da estrutura do *workflow* quando este está sendo executado em um ambiente remoto. Assim, o objetivo geral desta dissertação é oferecer apoio à adaptação dinâmica de *workflows* em tempo de execução. Esse apoio ao cientista visa a realização de mudanças estruturais dinâmicas no *workflow*, ou seja, ao longo de sua execução. Este objetivo é justificado pela hipótese de que, permitindo alterações dinâmicas, o tempo gasto para a obtenção dos resultados desejados seja minimizado como observado por Dias *et al.* (2011). Consequentemente, o tempo de execução e o custo financeiro atrelados ao experimento também serão reduzidos. O DynAdapt utiliza uma função de custo ponderada e uma base de proveniência em tempo real para escolher a alteração mais adequada a ser realizada no *workflow* atendendo aos critérios definidos pelo cientista. A função de custo pode considerar diversos critérios, tais como o tempo de execução, questões de qualidade e custo financeiro. Toda informação usada no cálculo desta função de custo pode ser consultada em tempo de execução no repositório de proveniência. Alternativamente, o DynAdapt também permite que o cientista faça alterações arbitrárias no *workflow*, trocando uma atividade por outra equivalente. A contribuição desta dissertação, por meio do desenvolvimento de DynAdapt, é mostrar a viabilidade e o benefício potencial do uso de mudanças dinâmicas na especificação de workflows científicos. DynAdapt foi implementado (Santos *et al.* 2013a) e avaliado experimentalmente (Santos *et al.* 2013b) com workflows de aplicações reais em bioinformática, evidenciando os ganhos com a troca de atividades equivalentes.

O restante dessa dissertação está organizado da seguinte forma: no Capítulo 2 são apresentados aspectos de *workflows* dinâmicos e uma revisão da literatura relacionada; no Capítulo 3 é apresentada a abordagem proposta desta dissertação; No capítulo 4 apresentamos os experimentos realizados para avaliação da abordagem proposta e os seus resultados; finalmente, no Capítulo 5 apresentamos as conclusões deste trabalho e algumas sugestões de trabalhos futuros.

2. Aspectos Dinâmicos na Execução de *Workflows*

Análises científicas são naturalmente dinâmicas, uma vez que cientistas exploram possibilidades ao trabalhar com diferentes hipóteses que são avaliadas ao longo do ciclo de vida do experimento. O ciclo de vida do experimento científico é composto basicamente pelas etapas de composição, execução e análise, que ocorrem progressivamente repetidas vezes até que o resultado desejado seja alcançado, conforme mostrado na Figura 1 e definido por Mattoso *et al.* (2010). Neste ciclo, os dados de proveniência coletados, armazenados e consultados ao longo de todas as etapas tem papel fundamental na condução do experimento. A proveniência é toda informação da história do *workflow*, o seu *pedigree* (Freire *et al.* 2008). As informações de proveniência podem ser divididas em duas categorias, a proveniência prospectiva e a proveniência retrospectiva. A proveniência prospectiva é toda informação sobre a especificação do *workflow*, como atividades e dependências de dados. Já a proveniência retrospectiva é toda informação acerca da execução do *workflow*, como instantes de início e fim da execução, quais atividades foram executadas, quem executou o *workflow*. As informações de proveniência são fundamentais para a validação e reprodutibilidade de um experimento.

Na etapa de composição, o cientista constrói o workflow definindo as atividades que serão utilizadas e suas relações de dependência. Uma vez que o workflow esteja montado, o cientista parte para a fase de execução, onde o workflow é executado e são gerados resultados. Na fase de análise, o cientista avalia os resultados gerados e, então, de acordo com a análise, parte novamente para a fase de composição para realizar os ajustes no workflow a fim de obter novos resultados. Em geral, este ciclo é executado iterativamente até que os resultados desejados sejam obtidos. Entretanto, não existem fronteiras rígidas entre as três fases do ciclo, ou seja, o cientista não precisaria terminar completamente a composição para iniciar a execução do *workflow*, nem aguardar o fim da execução para começar a analisar os resultados.

Durante a execução de um *workflow*, o cientista poderia fazer uma análise parcial dos resultados e, baseado nesta análise, realizar ajustes no *workflow* para aprimorar os resultados. Entretanto, de acordo com o levantamento da literatura em workflows científicos, não foram encontradas soluções ou modelos que permitissem esse tipo de mudança dinâmica, o que motivou o desenvolvimento nesta dissertação. DynAdapt oferece apoio ao cientista principalmente na fase de execução, permitindo

que o cientista possa conduzir dinamicamente a execução do workflow através da troca dinâmica de atividades. Um recurso previsto pelo DynAdapt é um mecanismo de consultas às informações de proveniência em tempo de execução. Assim, durante a etapa de execução do workflow, o cientista pode analisar os dados parciais e decidir qual ajuste será feito em tempo de execução.

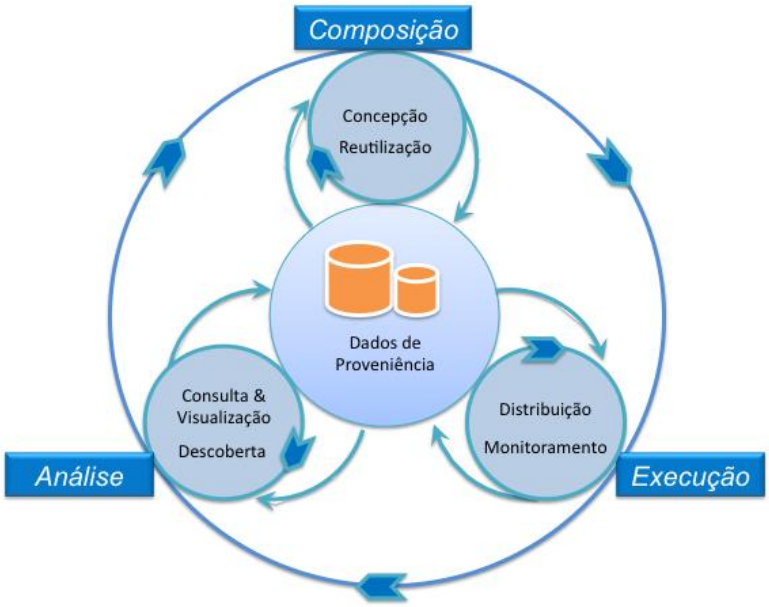

Figura 1 Ciclo de vida do experimento científico adaptado de (Mattoso *et al.* 2010)

O conceito de *workflows* dinâmicos (Gil *et al.* 2007) surgiu para apoiar as características dinâmicas do experimento e a natureza exploratória da ciência. Neste capítulo, apresentamos características de *workflows* dinâmicos e destacamos alguns trabalhos relacionados.

2.1 *Workflow* dinâmico

Atualmente os ambientes de desenvolvimento e execução de experimentos científicos podem ser distribuídos geograficamente. Cientistas utilizam equipamentos instalados em diferentes centros de pesquisa para trabalhar no desenvolvimento de um mesmo experimento. Em função do grande volume de dados a ser explorado, a execução de tal experimento também pode requerer recursos computacionais distribuídos como grades computacionais (Foster e Kesselman 2004), *clusters* (Dantas 2005) e nuvens computacionais (Vaquero *et al.* 2009). O apoio a *workflows* dinâmicos, adaptativos e conduzidos pelo cientista visa possibilitar e acelerar a metodologia científica distribuída e colaborativa por meio de rápida reutilização, melhoria e

exploração de modelos acompanhados de contínua adaptação (Gil *et al.* 2007). O desafio é construir mecanismos para compor e gerenciar *workflows* dinâmicos a fim de possibilitar a reutilização e reprodução de resultados significantes em experimentos.

Um cenário comum em experimentos científicos é o uso de resultados obtidos em uma execução de um *workflow* inicial para a tomada de decisões acerca de uma nova análise mais complexa realizada em seguida. *Workflows* científicos podem ser compostos de forma dinâmica para analisar resultados parciais antes de serem tomadas decisões para dar continuidade aos passos seguintes da análise. As decisões a respeito do experimento também podem ser desencadeadas a partir de eventos externos, que podem alterar a estrutura e a semântica dos *workflows* de forma que, dependendo do evento ocorrido, o *workflow* pode apresentar um comportamento diferente.

Em experimentos de larga escala, os programas executados são demorados e em muitos casos requerem intervenção humana e condução dinâmica ao longo do processo experimental. O apoio ao cientista na condução de experimentos que envolvam *workflows* dinâmicos é um desafio. Neste cenário, dada a complexidade dos *workflows*, faz-se necessário um sistema de apoio à tomada de decisões para acomodar as necessidades dos cientistas ao buscar informações para compreender os processos envolvidos em todas as etapas do experimento. É preciso também um mecanismo eficiente de consultas que permita que o cientista faça buscas e compreenda as informações de proveniência. Baseado nos requisitos destacados por Gil *et al.* (2007) para possibilitar o desenvolvimento de *workflows* dinâmicos, podemos destacar os seguintes requisitos: (i) desenvolvimento colaborativo e distribuído geograficamente; (ii) sistema de consultas eficiente aos resultados e dados de proveniência para análises sofisticadas dos dados obtidos na execução; (iii) adaptação do *workflow* em resposta a modificações nos dados de entrada, na estrutura ou na configuração do workflow; e (iv) exploração do espaço de parâmetros (dados de entrada) em fatias, de forma que os cientistas possam tomar decisões tais como descartar dados de entrada ou comparar resultados obtidos com diferentes fatias.

No domínio da bioinformática, cientistas normalmente precisam consultar e analisar resultados parciais e ajustar atividades e parâmetros do *workflow* durante a execução. Nesta área podemos citar exemplos de *workflows* que requerem computação de alto desempenho (CAD), como o SciPhy (Ocaña *et al.* 2011b), o SciPhylomics (Oliveira *et al.* 2012a), o SciHmm (Ocaña *et al.* 2011a) e o SciEvol (Ocaña *et al.* 2012).

Uma característica comum dentre estes *workflows* é a necessidade de execuções paralelas com manipulação intensiva de dados. Nestes *workflows*, o cientista não sabe *a priori* qual o programa mais adequado para ser usado para um determinado conjunto de dados de entrada. Assim os cientistas precisam explorar diferentes configurações do *workflow* até encontrar o resultado desejado.

O trabalho de Gil *et al.* ressalta a natureza dinâmica dos *workflows* evidenciando sua evolução ao longo do ciclo de vida do experimento. Segundo os autores é necessário que os SGWfC deem suporte às alterações dinâmicas que ocorrem nos experimentos, sejam elas alterações de ambiente, na estrutura ou nos requisitos. O trabalho apresentado nesta dissertação visa atender o desafio de apoiar alterações dinâmicas na estrutura de *workflows* científicos em tempo de execução. O foco do trabalho apresentado nesta dissertação é apoiar o desenvolvimento de *workflows* dinâmicos por meio da adaptação dinâmica do *workflow* de acordo com critérios definidos pelo cientista. Ou seja, permitir alterações na definição do *workflow*, em tempo de execução, de maneira que estas alterações sejam refletidas nas instâncias em execução dinamicamente.

2.2 Trabalhos Relacionados

Gil *et al.* 2007 fazem uma análise sobre os principais desafios a serem alcançados no contexto dos *workflows* científicos. Os autores listam uma série de requisitos que, segundo eles, deverão ser atendidos pelos SGWfC, de acordo com a evolução e escalabilidade dos experimentos. Dentre os requisitos enumerados no texto, podem-se encontrar os requisitos relacionados às características dinâmicas dos *workflows* científicos. Mattoso *et al.* (2013) fazem uma análise do estado da arte e direções futuras sobre a condução dinâmica da execução de *workflows* pelo cientista. O trabalho propõe três questões principais relacionadas à condução do *workflow* pelo cientista: monitoramento da execução, análise de dados em tempo de execução e interferência dinâmica na execução do *workflow*. Segundo o texto, a interferência dinâmica é a principal etapa da condução do *workflow* conduzido pelo usuário. A análise dos resultados parciais em tempo de execução e interferência dinâmica do cientista são as principais questões tratadas no trabalho apresentado nesta dissertação. Dentre as questões em aberto, ressaltadas por Mattoso *et al.*, podemos destacar o apoio à decisão e a máquina de *workflow* dinâmico, que são tratadas nesta dissertação.

Podemos separar as abordagens de apoio a *workflows* dinâmicos em três categorias: abordagens de monitoramento, de análise em tempo de execução e de intervenção dinâmica. A seguir analisamos as principais abordagens existentes para cada uma dessas categorias.

2.2.1 Monitoramento da Execução

O monitoramento da execução do *workflow* é um recurso importante em *workflows* dinâmicos para apoiar a condução dinâmica pelo cientista. Este monitoramento dá ao cientista a consciência do que está ocorrendo, permitindo que ele acompanhe a execução do experimento. A maioria dos SGWfCs possui recursos para o monitoramento da execução do *workflow*. Entretanto, cada SGWfC dá este suporte em diferentes granularidades. Emeakaroha *et al.* (2011) oferecem um arcabouço para monitorar a infraestrutura de execução do *workflow* na nuvem enquanto as aplicações são executadas. Mas não permitem a condução do *workflow* pelo usuário. Os resultados deste monitoramento podem ser usados para otimização da execução do *workflow*, escalonando atividades para outras máquinas virtuais. Lee *et al.* (2007) e Truong et al. (2006) apresentam um mecanismo para apoio *online* ao monitoramento do desempenho e análise de *workflows* orientados a serviço. Este monitoramento é interessante na medida em que oferece base para identificação de problemas na execução do *workflow*, permitindo que os cientistas possam tomar ações corretivas. Vahi *et al.* (2012) apresentam o Stampede, uma infraestrutura de monitoramento que foi integrada aos SGWfCs Pegasus e Triana. O ideia é prover um monitoramento genérico em tempo real sobre mais de um SGWfC concorrentemente. Os resultados mostraram que o Stampede pode monitorar execuções de *workflows* em múltiplos e diferentes SGWfCs. O Stampede tem foco em *workflows* de computação de alto desempenho, o que requer que o cientista acompanhe a execução em uma estação de trabalho. Esta solução pode não ser interessante para execuções muito longas, uma vez que o cientista deverá acompanhar toda a execução do workflow. Missier *et al.* (2010) propõem a criação de gatilhos para checar por problemas no *workflow* e emitir alertas para o cientista.

Estas abordagens citadas acima foram projetadas para a otimização no tratamento de exceções e figuram como um importante passo no apoio à condução do *workflow* pelo cientista. Todavia, estas abordagens não permitem a intervenção do cientista na execução nem alterações dinâmicas no *workflow* em tempo de execução. O cientista deve estar a par dos dados relacionados ao *workflow* ao longo de toda sua

execução. Ele deve poder, a qualquer instante, consultar os dados disponíveis para decidir por analisar resultados preliminares ou interferir na execução. Abordagens para condução do *workflow* pelo usuário devem evitar que o monitoramento exija que o usuário tenha que ficar diante de uma estação de trabalho durante toda a execução do *workflow*. É preferível uma abordagem como o SciLightning (Pintas *et al.* 2013) que notifica o cientista sobre os eventos mais importantes por meio de dispositivos móveis e mensagens em redes sociais. Com o SciLightning, o cientista pode configurar eventos relacionados à execução das atividades e aos dados gerados. O SciLightning provê ainda apoio para consulta à proveniência e a resultados parciais.

2.2.2 Análise de Dados em Tempo de Execução

A análise dos dados gerados durante a execução do *workflow* também é um recurso importante para a condução dinâmica do experimento pelo cientista. Em geral, alguns resultados parciais já são suficientes para uma análise em tempo de execução. Os cientistas precisam de ferramentas para capturar os dados desejados, consolidá-los e analisá-los. Os resultados visualizados podem ser enriquecidos com informações de proveniência e analisados com ferramentas de estatística, por exemplo, a fim de facilitar a tomada de decisão sobre o experimento em execução.

Algumas abordagens oferecem recursos de visualização e análise de dados após a completa execução do *workflow*. O VisTrails (Callahan *et al.* 2006) oferece diversos recursos para visualizar e comparar resultados de diferentes execuções do *workflow*. Entretanto, o VisTrails não provê recursos para execução em ambientes de computação de alto desempenho nem acesso às informações de proveniência em tempo de execução. O cientista também pode conectar a sua ferramenta de execução de *workflow* à plataforma de visualização do ParaView (Fabian *et al.* 2011). O cientista pode instrumentar o seu *workflow* para exportar os resultados produzidos para o ParaView. Esta abordagem, por ser desacoplada do SGWfC, também não configura como um recurso de acesso a informações de proveniência em tempo de execução, pois depende da exportação dos dados por parte do SGWfC. Como pode-se observar, nenhuma dessas abordagens atende ao desafio de workflow dinâmico no que diz respeito a análise da proveniência em tempo de execução.

Para oferecer apoio à análise em tempo de execução, os SGWfC precisam prover acesso às informações de proveniência e aos resultados do experimento para o cientista enquanto o experimento está sendo executado. Apenas os sistemas

SciCumulus (Oliveira *et al.* 2010) e Chiron (Ogasawara *et al.* 2013) oferecem esse tipo de análise em tempo de execução. Está em andamento, junto ao SciCumulus, o sistema Prov-Vis (Horta *et al.* 2013), que permite ao cientista submeter consultas à base de proveniência do workflow sendo executado em ambientes de nuvens computacionais e prover a visualização de resultados parciais do workflow científico em paredes de monitores de visualização (*tiled wall display*).

2.2.3 Interação Dinâmica

A interação dinâmica é a principal etapa na condução dinâmica de *workflows* pelo cientista. Para isso, é preciso que o cientista possa visualizar o que está ocorrendo na execução do *workflow*. Desta forma, baseado no monitoramento da execução e na análise dos dados em tempo de execução, o cientista pode decidir se deve interagir com a execução. Esta interação pode ser simples como parar o *workflow*, parar uma atividade ou reexecutar atividades, por exemplo. Ou pode ser mais complexa, como a troca de uma atividade, alteração na estrutura do *workflow* ou mudanças nos parâmetros, por exemplo. No nosso exemplo de análise filogenética, o cientista pode ter interesse em trocar o método de AMS, valores dos parâmetros ou parte dos dados de entrada. Quanto ao nível do apoio à interação do cientista na execução do *workflow*, os SGWfCs existentes variam desde nenhum apoio até a reexecução automática de um *workflow* quando uma falha é detectada.

Bowers *et al.* (2006) construíram um dos primeiros mecanismos de interação na execução de *workflows* no SGWfC Kepler. Esta abordagem torna o *workflow* científico mais robusto por meio da implementação de uma estratégia de tolerância a falhas no fluxo de dados. A principal ideia é usar dados de proveniência para reexecutar atividades que falharam ou iniciar a execução de uma nova atividade. Missier *et al.* (2010) propuseram o uso de gatilhos (*triggers)* , muito comum em Sistemas de Bancos de Dados, em *workflows* científicos para reagirem a eventos no Taverna *Workbench*. Qualquer ação no Taverna *Workbench* pode desencadear uma ação automática, que pode ser reexecutar atividades ou parar a execução corrente. Samak *et al.* (2011) propõem um arcabouço para tratamento de falhas em *workflows* executados no SGWfC Pegasus. São aplicadas árvores de regressão para classificar se uma atividade foi executada com sucesso ou falha. O arcabouço aprende o comportamento da execução da atividade e prevê futuras falhas. Estes trabalhos oferecem apoio a recursos importantes para alterações dinâmicas em *workflows*. Entretanto todos estes são focados em

tolerância a falhas. E diferentemente do trabalho proposto nesta dissertação as ações de contingência precisariam ser programadas *a priori*, o que pode ser inviável para *workflows* executados em ambientes de computação de alto desempenho. Além disso, estes trabalhos são dependentes de SGWfC. Seria interessante uma abordagem mais genérica e independente de SGWfC, pois isto permitiria que outras ferramentas pudessem usufruir dos benefícios providos pela abordagem.

No nosso grupo de pesquisas, algumas iniciativas (Dias *et al.* (2011), Ocaña et al. (2011a), Costa *et al.* (2012b)) foram desenvolvidas no sentido de prover interação dinâmica durante a execução de workflows científicos. Muitas serviram de motivação e base para a proposta e o desenvolvimento desta dissertação. A seguir, analisamos cada uma delas.

Dias *et al.* (2011) apresentam uma abordagem para varredura dinâmica de parâmetros em *workflows* executados em ambientes de computação de alto desempenho. Os autores propõem uma estrutura para permitir ajustes dinâmicos na exploração de parâmetros do *workflow* em tempo de execução. Baseado em resultados parciais em tempo de execução o cientista pode decidir quando interromper uma iteração de um *workflow* científico e fazer ajustes nos parâmetros utilizados dinamicamente. Apesar de prover apoio à interferência dinâmica do cientista na execução do *workflow*, a abordagem proposta por Dias *et al.* 2011 não trata de alterações dinâmicas nas atividades do mesmo, como é o objetivo desta dissertação.

Costa *et al.* (2012b) mostraram que o acesso às informações de proveniência em tempo de execução permitem uma rápida identificação de falhas na execução do *workflow*, o que ajuda na solução de problemas sem interferir no desempenho do experimento. Em seguida, Costa *et al.* (2012a) propuseram um conjunto de heurísticas baseadas nas informações de proveniência em tempo de execução para detecção de falhas em atividades para gerenciar suas reexecuções em nuvens computacionais. Costa *et al.* (2012b) toma vantagem da proveniência em tempo de execução para prover tolerância a falhas e ser independente de SGWfC, considerando o modelo PROV (Moreau *et al.* 2011), recomendado a padrão de proveniência.

Ocaña et al. (2011a) propuseram um *workflow*, o SciHmm, para, utilizando cadeias de Markov ocultas (Eddy 1996), escolher o algoritmo de alinhamento genético mais adequado baseado em informações de proveniência. A solução adotada não trata do problema de alteração dinâmica, mas evidencia a possibilidade de, a partir dos dados

de proveniência, uma determinada implementação de uma atividade ser mais adequada do que outras. O trabalho proposto nesta dissertação se aproxima do trabalho de Ocaña *et al.* na medida em que propõe uma abordagem para, baseado em informações de proveniência, eleger dinamicamente a implementação mais adequada de uma atividade e fazer sua substituição em tempo de execução. Vale lembrar que as implementações alternativas devem estar definidas previamente. A principal diferença é que Ocaña *et al.* propõem o *workflow* SciHmm para eleger a atividade mais adequada baseado nas informações de proveniência, porém não tratam da troca dinâmica em tempo de execução nem da condução do experimento dinamicamente pelo cientista. Além disso, diferentemente da abordagem proposta nesta dissertação, onde os critérios utilizados na troca de atividades são controlados pelo cientista, o critério de escolha na abordagem de Ocaña *el al.* está definido dentro da atividade.

Todas as abordagens mencionadas anteriormente oferecem apoio importante para os aspectos dinâmicos na execução de *workflows*. Entretanto, não oferecem apoio para a troca de atividades do *workflow* quando este já se encontra em execução. Das abordagens estudadas para a construção da abordagem proposta nessa dissertação, apenas as abordagens de *workflows* de negócio oferecem apoio similar a este tipo de alteração na estrutura do *workflow*. Embora não adequadas para gerenciar a execução de workflows científicos, analisamos as soluções da área de negócios visando a eventuais adaptações para a área científica.

Kammer *et al.* (2000) propõem apoio dinâmico por meio da utilização de técnicas de tratamento de exceção. Kammer *et al.* discutem tipos de exceções ressaltando os impactos que elas podem causar no workflow. Baseados nesta discussão, Kammer *et al.* sugerem pontos a serem tratados no que diz respeito à adaptação do workflow, incluindo estratégias para evitar exceções, detectá-las se for o caso, e tratá-las. Os autores apresentam características como aspectos de workflows dinâmicos são tratados no sistema de workflows Endeavors. O Endeavors usa um modelo de objetos em camadas para definição e especificação de artefatos, atividades e recursos envolvidos em um processo. As atividades são executadas por interpretadores, que percorrem as atividades e enviam comandos para invocar algum comportamento específico dos objetos. Através de uma interface, o Endeavors permite que o usuário especifique dinamicamente os interpretadores e controle a execução. Ellis *et al.* (1995) utiliza redes de Petri (Hoheisel e Alt 2007) para analisar alterações na estrutura do

workflow. Ellis et al definem duas regiões de mudança, a região antiga, que compreende todos os elementos do workflow que sofrerão alguma mudança, e a nova região, que é composta pelas alterações a serem feitas no workflow. Ellis et al definem, como exemplo, uma classe de mudança chamada *synthetic cut-over change*, onde antiga e nova região são mantidas de forma que as execuções da antiga região que já foram iniciadas continuem sua execução. Ellis et al aplicam um formalismo para provar que esta classe de alteração mantém a corretude quando ocorre a situação em que uma execução da nova região é também uma execução elementar da antiga região, ou seja, a nova região faz menos do que a antiga região. Todas estas abordagens foram implementadas para os cenários de *workflow* de negócio, que são centrados no controle das atividades. São construídos para atividades de negócio, não para programas científicos, que são tipicamente centrados nos dados. Além disso, estes trabalhos não consideram diversos fatores importantes para os cientistas como proveniência dos dados, desempenho, qualidade dos resultados do experimento e custo financeiro decorrente do ambiente de execução do *workflow*.

A abordagem proposta nesta dissertação é voltada para *workflows* científicos centrados em dados. Ela tem foco nos benefícios que as alterações dinâmicas podem trazer ao processo científico. Baseado em critérios definidos pelo cientista, o DynAdapt altera as atividades que estão em execução. A Tabela 1 apresenta um resumo dos trabalhos de workflow científico relacionados, agrupados pelos aspectos de monitoramento de execução, análise de proveniência em tempo de execução e interação dinâmica.

Tabela 1 Resumo dos principais trabalhos relacionados.

Monitoramento da Execução	
Emeakahora *et al*., 2011	Monitoramento da infraestrutura
Lee *et al*., 2007 e Truong *et al*., 2006	Monitoramento do desempenho e análise
Vahi *et al*., 2012	Monitoramento genérico em tempo real para múltiplos SGWfCs
Missier *et al*., 2010	Gatilhos para checar por problemas
Análise da proveniência em tempo de execução	
Callahan *et al*., 2006	Visualização e comparação de resultados
Fabian *et al*., 2011	Plataforma de visualização
Oliveira *et al*., 2010	Consultas analíticas na nuvem
Ogasawara *et al*., 2013	Consultas analíticas em cluster
Horta *et al.*, 2013	Visualização e consultas à base de proveniência em tempo de execução
Interação dinâmica	
Bowers *et al*., 2006	Estratégia de tolerância a falhas no Kepler
Missier *et al*., 2010	Reação a eventos por meio de *triggers*
Samak *et al*., 2011	Arcabouço para tratamento de falhas no SGWfC Pegasus
Dias *et al*., 2011	Varredura dinâmica de parâmetros
Ocaña *et al*., 2011	Uso de proveniência para interromper execução
Costa *et al*., 2012	Proveniência em tempo de execução para tratamento de falhas

3. DynAdapt - mudanças dinâmicas na especificação de workflows científicos

No contexto de *workflows* científicos podem ocorrer casos em que uma mesma atividade do *workflow* possua várias implementações correspondentes, de forma que estes programas podem ser vistos como alternativos entre si. De acordo com variáveis como ambiente de execução ou requisitos do experimento, o uso de um programa pode ser mais vantajoso que outro em termos de tempo de execução, qualidade dos resultados, dentre outros aspectos. Geralmente, após o término de uma primeira execução, o cientista analisa os resultados obtidos e pode então decidir qual implementação deverá ser utilizada em uma atividade para que se atinja o resultado desejado. Em seguida, ele executa todo o *workflow* novamente, porém utilizando a nova implementação escolhida para a atividade.

Em geral, de acordo com resultados parciais de uma execução já é possível inferir qual a implementação que melhor atende ao cientista em relação a um determinado critério, como qualidade dos dados gerados, por exemplo. Assim, o objetivo do DynAdapt é permitir que, baseado na análise da proveniência em tempo de

execução, o cientista possa realizar adaptações nas atividades do *workflow* dinamicamente, sem que seja necessário reiniciar a execução do mesmo. Desta forma, espera-se que o tempo gasto para atingir o resultado desejado para o experimento seja reduzido.

O DynAdapt é uma proposta, independente de SGWfC, para apoiar o cientista na troca automática da implementação das atividades do *workflow*, de acordo com os objetivos definidos pelo cientista antes da execução do experimento. Com o DynAdapt, o cientista conduz a execução do *workflow* de acordo com os seus objetivos que podem variar desde o tempo de execução até a qualidade dos resultados. Assim, não é necessário reexecutar o *workflow* todo novamente para que as alterações sejam consideradas.

O DynAdapt provê apoio a alterações dinâmicas na estrutura dos workflows científicos, utilizando proveniência em tempo de execução. Por meio de um modelo ponderado de custo, a abordagem escolhe automaticamente o programa mais adequado para uma atividade específica. Além disso, o cientista pode interferir na execução do workflow e escolher outra atividade para ser executada caso não esteja satisfeito com a escolha automática do DynAdapt. A Figura 2 apresenta uma visão geral de uma abordagem para permitir a condução dinâmica do experimento pelo usuário. A ideia é que o cientista possa monitorar e interagir com a máquina de workflow enquanto este é executado.

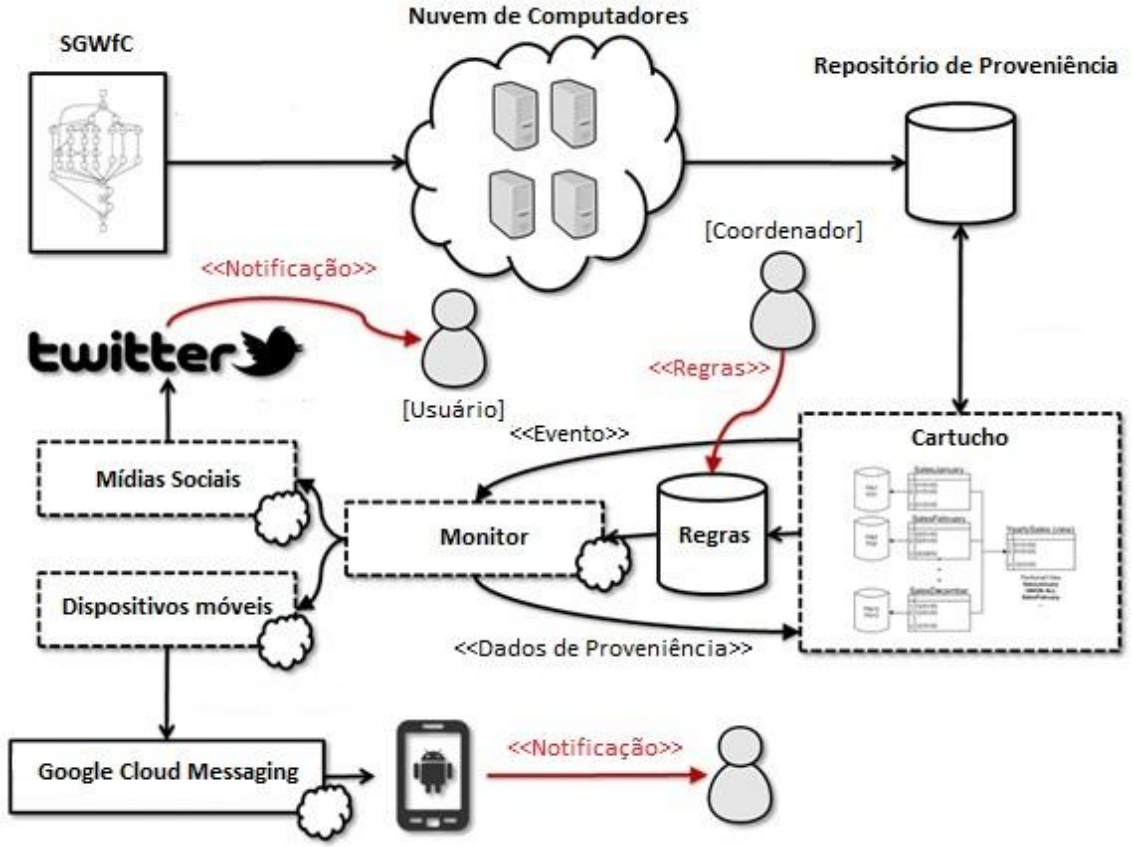

Figura 2 Visão gerar da abordagem proposta

O DynAdapt consiste na implementação de quatro componentes construídos para permitir ao cientista conduzir o experimento através da interação com o ambiente de execução: (i) o Agente de Composição, que é o responsável por definir os parâmetros do DynAdapt e as implementações alternativas das atividades do *workflow*; (ii) O Repositório de Proveniência, que contém informações sobre a estrutura do *workflow* e informações sobre suas execuções, que podem ser utilizadas como base para alterações estruturais; (iii) o Agente de Adaptação que é o responsável por fazer uma interface com a máquina de *workflow* para realizar as alterações na estrutura; e (iv) o Rendez-vous que é o responsável por prover informações de proveniência em tempo de execução para o Agente de Adaptação quando este não é oferecido nativamente pelo SGWfC. A ideia é que o DynAdapt possa ser integrado a componentes de notificação e monitoramento, como os apresentados na Figura 2 para permitir ao cientista interferir na execução do experimento.

Ao longo deste capítulo apresentamos, inicialmente, um formalismo para a representação de *workflows* científicos e o modelo ponderado de custo utilizado para a eleição da variabilidade mais adequada de determinada atividade.

3.1 Formalismo e Modelo de Custo Ponderado

Para ajudar na escolha da implementação mais adequada de uma determinada atividade, o DynAdapt faz uso de um modelo de custo ponderado, que associa a cada programa alternativo, um conjunto de fatores a serem considerados quando houver alguma alteração nos critérios de execução definidos pelo cientista. Tempo de execução, custo financeiro e qualidade dos resultados são exemplos de fatores que podem estar associados a um conjunto específico de programas alternativos entre si. Baseado em tais fatores, o DynAdapt escolhe a melhor implementação de acordo com um dado critério definido pelo cientista. Para apresentar o modelo de custo ponderado, primeiramente definimos uma base formal para representação do *workflow*. Para esta base formal, estendemos a representação algébrica de workflows definida por Ogasawara *et al.* (2011), por definir operações baseadas na álgebra relacional e por adotar relações para representar as atividades do workflow. Essa representação estruturada facilita a modelagem de variabilidades para as atividades.

Um *workflow* científico é definido por Ogasawara *et al.* (2011) como um Grafo Direcionado Acíclico $W(Y, Dep)$. O conjunto Y de vértices em W contém todas as atividades do *workflow* a serem executadas (potencialmente em paralelo) e as arestas *Dep* representam as dependências de dados entre duas atividades em W. Toda atividade $Y_i \in Y$ do *workflow* consome um conjunto $R=\{R_{i1},..., R_{in}\}$ de relações com esquema $\{\Re_{i1}, ..., \Re_{in}\}$ e produz uma relação de saída T com esquema $\Im$. Nesta dissertação, adaptamos o modelo de Ogasawara *et al.* (2011) para permitir mudanças dinâmicas na definição do workflow, assim, cada atividade passou a ser associada a um conjunto de variabilidades, ou seja $Y_i \in \{V_1, ..., V_k\}$. Logo, para cada relação $R_{ij}(\Re_{ij})$ de entrada de Y_i, $\forall\ V_x \in \{V_1, ..., V_k\}$ possui uma relação entrada $R_{iv}(\Re_{iv})$ tal que $\Re_{iv} \subseteq \Re_{ij}$. Ou seja, toda a variabilidade de uma atividade Y_i possui o mesmo conjunto de relações de entrada. Todas as variabilidades de uma atividade devem estar preparadas para receber o mesmo conjunto de relações de entrada.

A fim de facilitar a automatização do processo de escolha da atividade mais adequada, o DynAdapt utiliza um modelo de custo ponderado para fazer a associação de

um conjunto de atividades alternativas e um conjunto de fatores a serem considerados no momento da escolha da atividade que vai ser executada. Este modelo de custo é baseado nos trabalhos de Boeres *et al.* (2011) e Oliveira *et al.* (2012a) e é projetado para permitir que os fatores de um conjunto de atividades possam ser ponderados. Cada variabilidade de uma atividade possui v_1, ..., v_n valores referentes aos n fatores com pesos w_1, ..., w_n tal que $\sum_{i=1}^{n} wi = 1$. A média ponderada dos fatores de uma atividade é $w_1.v_1 + w_2.v_2 + ... + w_n.v_n$.

Como para cada fator é atribuído um peso, quanto maior o seu peso, mais valorizado é o fator no cálculo da média ponderada. Para uma mesma atividade Y_i, cada variabilidade pode ter valores diferentes para seus fatores, mas os pesos de cada fator são iguais para cada variabilidade. Por exemplo, consideremos os fatores desempenho e qualidade. Para uma atividade, podemos ter uma variabilidade com bom desempenho, porém com uma qualidade inferior nos dados gerados. Ao mesmo tempo a mesma atividade pode ter outra variabilidade cujos dados gerados tem alto grau de qualidade enquanto possui desempenho inferior. Ou seja, elas possuem valores diferentes para os fatores desempenho e qualidade. Se o cientista necessita de mais qualidade, por exemplo, basta aumentar o peso deste fator. A ideia é que estes pesos possam ser atribuídos pelo cientista tanto antes da execução do *workflow* quanto durante. Seguindo essas informações sobre os fatores e pesos, o DynAdapt elege a implementação mais adequada para executar uma determinada atividade.

3.2 Agente de Composição

O agente de composição é o componente responsável por fornecer ao cientista uma interface para a definição do *workflow* e para o ajuste das informações do modelo de custo ponderado. A ideia é que o cientista possa, com esta interface, conduzir a execução do *workflow* por meio de ajustes nos pesos e valores dos fatores. A partir destes ajustes o agente de adaptação, que será apresentado em detalhes nas próximas seções, se encarrega de decidir, de acordo com o modelo de custo ponderado, as atividades mais adequadas para serem executadas. Alternativamente, o cientista pode escolher arbitrariamente uma implementação da atividade para substituir outra que esteja escalonada para execução. Esta interface funciona como um painel de controle para o cientista. Sempre que ele mudar o objetivo em relação à execução do *workflow*, basta acessar este componente e ajustar os pesos dos fatores considerados na execução do *workflow*. Uma vez que as informações de peso e fatores tenham sido alteradas, o

agente de adaptação é automaticamente invocado para realizar as alterações no *workflow* visando atender à vontade do cientista.

O Agente de Composição é executado uma vez antes do início da execução do *workflow*. Nesta etapa, o Agente de Composição interpreta um arquivo XML contendo todos os metadados associados às possíveis alterações estruturais, tais como as implementações alternativas e os fatores do modelo ponderado de custo para cada atividade. Existem duas maneiras de implementar este mecanismo: (i) utilizar um arquivo XML de configuração adicional além da especificação do *workflow* ou (ii) estender o esquema de definição do *workflow*. A primeira solução é não intrusiva e requer que um arquivo XML adicional seja submetido com o arquivo de definição do *workflow*. A estrutura do arquivo XML é apresentada na Figura 3, onde um workflow possui um conjunto de atividades. Cada atividade possui, dentre outros atributos, um conjunto de implementações alternativas. O DynAdapt guarda uma referência para o SGWfC a ser utilizado, o *workflow* a ser executado e suas atividades. Para cada atividade, o cientista define uma ou mais implementações alternativas, se necessário. O cientista deve definir a linha de comando do programa e o seu caminho para ser executado. Já a segunda solução é intrusiva e requer alterações no arquivo de especificação do SGWfC. Esta solução só é viável se o cientista puder ajustar a estrutura do arquivo de especificação do *workflow*.

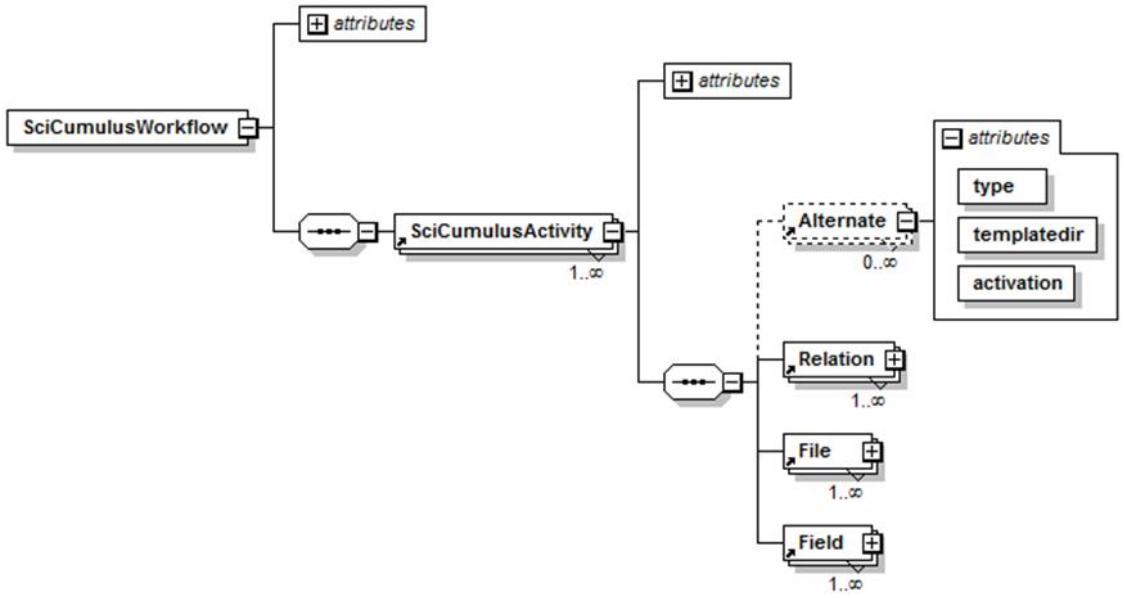

Figura 3 Esquema XML de configuração do DynAdapt.

Vale ressaltar que não é necessário que o cientista mantenha uma seção aberta do agente de composição durante toda a execução do *workflow*, que pode levar dias ou

até semanas. Alertas podem ser enviados de modo assíncrono para o cientista de acordo com o estado da execução do *workflow*. Após receber o alerta e analisar resultados parciais, o cientista pode utilizar o Agente de Composição igualmente de maneira assíncrona para requisitar a alteração de uma atividade do workflow.

Para esta dissertação não foi construído um Agente de Composição. O XML de configuração utilizado neste trabalho foi montado manualmente segundo o esquema apresentado na Figura 3 e as alterações no modelo de custo ponderado foram feitas diretamente na base de proveniência.

3.3 Agente de Adaptação

O Agente de Adaptação é o componente responsável por consultar os valores dos fatores, seus pesos e eleger qual implementação da atividade deve ser executada. Uma vez escolhida a implementação, o agente de adaptação altera as instâncias que estão em execução. Para isso é necessário que o agente de adaptação tenha permissão para alterar o plano de execução da máquina de *workflow*, para que as alterações sejam consideradas no *workflow* que está sendo executado.

Como mencionado anteriormente, a ideia é que a execução do *workflow* seja conduzida de acordo com a vontade do cientista. Caso ele mude de critério e deseje que o *workflow* seja executado segundo um novo objetivo, o DynAdapt deve permitir que o *workflow* seja alterado e passe a ser executado segundo o novo objetivo do cientista. Sendo assim, para que a execução esteja sempre de acordo com o objetivo do cientista, o agente de adaptação é invocado sempre que um dos seguintes eventos ocorrer: (i) a inclusão ou atualização do *workflow* na base de proveniência; (ii) o cientista invoca explicitamente a troca da atividade por uma implementação específica arbitrariamente; (iii) a inclusão ou atualização em alguma informação no modelo de fatores. Se o cientista escolher uma implementação arbitrariamente, o agente de adaptação substitui a atividade em execução no plano de execução atual com a implementação escolhida. Caso contrário, para cada implementação alternativa, o agente de adaptação executa o seguinte cálculo:

$$C = \sum_{\forall i} w_i \times v_i$$

Onde w_i é o peso do fator i para o conjunto de atividades alternativas e v_i é o valor do fator i para a variabilidade em questão. É eleita a variabilidade que apresentar o maior

valor para *C*. É importante ressaltar que o DynAdapt utiliza uma abordagem algébrica para *workflows* centrados em dados (Ogasawara *et al.* 2011), onde uma atividade só pode ser substituída por uma alternativa, se a alternativa for compatível com a atividade corrente. O termo compatível aqui se refere a possuir as mesmas interfaces de entrada e saída, *i.e.* consumir os mesmos dados e parâmetros e produzir os mesmos dados e parâmetros. No contexto do SGWfC Chiron, utilizado no estudo de caso deste trabalho, a compatibilidade entre atividades é verificada no nível de esquema. Se uma atividade consome uma relação que segue o esquema $\mathcal{R}$ e produz uma relação que segue o esquema $\mathcal{I}$, a atividade alternativa deve seguir os esquemas $\mathcal{R}$ e $\mathcal{I}$ também. Embora a restrição apresentada pareça grande, na prática, se uma variabilidade *A* consome o esquema $\mathcal{A}$ e outra variabilidade *B* consome o esquema $\mathcal{B}$, basta garantir o esquema de entrada $\mathcal{R} \supseteq \mathcal{A} \cup \mathcal{B}$. Para o esquema de saída $\mathcal{I}$, a relação é análoga. Se as atividades seguintes do *workflow* não estiverem preparadas para consumir o esquema $\mathcal{I}$ (pois a relação poderá ter campos vazios), pode-se ter no *workflow* uma atividade que produza uma nova relação compatível com as atividades seguintes do *workflow* baseados nos dados de saída da relação com esquema $\mathcal{I}$.

Para realizar alterações no plano de execução do *workflow*, o DynAdapt precisa acessar as informações de proveniência do *workflow* em execução. Todavia, muitos SGWfCs não permitem consultas à base de proveniência em tempo de execução. Assim, o Agente de Proveniência figura como um componente opcional responsável por prover informações de proveniência em tempo real para o DynAdapt. Além disso, este componente interage com o Agente de Adaptação para trocar informações sobre as alterações feitas sobre o plano de execução do *workflow*.

Em nível de implementação, Agente de Adaptação consiste de um cartucho programado em Java que possui duas operações principais: (i) trocar uma atividade por uma variabilidade específica e (ii) trocar uma atividade pela variabilidade mais adequada de acordo com o modelo de custo ponderado. A primeira operação deve ser usada quando o cientista desejar explicitamente fazer a troca de uma atividade por uma determinada variabilidade independente de modelo de custo. Assim o Agente de Adaptação deve receber como parâmetro uma atividade e a variabilidade pela qual esta atividade deve ser substituída. Em seguida, o Agente de Adaptação aborta a execução da atividade que será substituída e cria novas ativações para a execução da variabilidade escolhida. Para esta operação poderia ser utilizada uma interface para que o cientista

possa informar a atividade e a variabilidade envolvidas. A segunda operação deve ser chamada sempre que houver alguma alteração no modelo de custo ponderado, como por exemplo, a alteração de algum peso. Mudanças no modelo de custo ponderado podem significar mudanças nos requisitos do experimento, o que pode requerer mudanças no plano de execução do workflow com a troca de uma atividade pela variabilidade mais adequada de acordo com os requisitos do experimento. Esta operação é executada automaticamente a cada alteração feita no modelo de custo ponderado. Uma vez que ocorra uma alteração nos fatores ou pesos do modelo de custo ponderado, o Agente de Adaptação calcula qual variabilidade é a mais adequada. Após a escolha da variabilidade, o Agente de Adaptação aborta as ativações da atividade que vai ser substituída e cria novas ativações para a variabilidade escolhida. Para esta dissertação estas duas operações foram construídas de modo a se comunicar com o repositório de proveniência do Chiron, onde o plano de execução do workflow é mantido. Para que estas operações sejam utilizadas em outras máquinas de workflow, é necessário a construção de cartuchos para manipulação dos planos de execução específicos de cada máquina.

3.4 Modelo de Proveniência

A arquitetura proposta no DynAdapt faz uso do repositório de proveniência como o seu componente central. Todos os outros componentes do DynAdapt interagem diretamente com o repositório de proveniência para executar suas ações. Todas as informações de proveniência prospectiva e retrospectiva do experimento são acessadas neste componente.

O modelo de proveniência adotado no repositório do DynAdapt é uma extensão do PROV-Wf (Costa *et al.* 2013), que é um modelo construído para acomodar informações de proveniência de diferentes SGWfC e permitir consultas em tempo de execução ao *workflow*. Este modelo é baseado no modelo de referência PROV (Moreau *et al.* 2011) da W3C, que permite a representação de entidades, agentes e atividades envolvidos na geração de um dado e seus relacionamentos. A extensão do PROV-Wf projetada para o repositório do DynAdapt consiste na inclusão de uma entidade para representar uma alternativa de uma atividade do *workflow*, chamada de variabilidade, inspirada nos conceitos linhas de produto de *software* (Pohl *et al.* 2005) e de linha de experimento (Ogasawara *et al.* 2009), e na inclusão do modelo de custo ponderado, descrito na Seção 3.1. A ideia de utilizar o conceito de variabilidade tem por objetivo a

representação de di ferentes implementações de uma determinada atividade sem alterar o fluxo do experimento. Toda a informação necessária para utilizar as rotinas de adaptação implementadas neste trabalho são obtidas seguindo o modelo *PROV-Wf*, apresentado na Figura 4. A extensão projetada sobre o PROV-Wf é apresentada na Figura 5. Os <<estereótipos>> indicam a representação de cada elemento na ontologia do PROV. O modelo de proveniência do DynAdapt pode ser separado do repositório de proveniência da máquina de *workflow*, apesar de existir relações entre ambos.

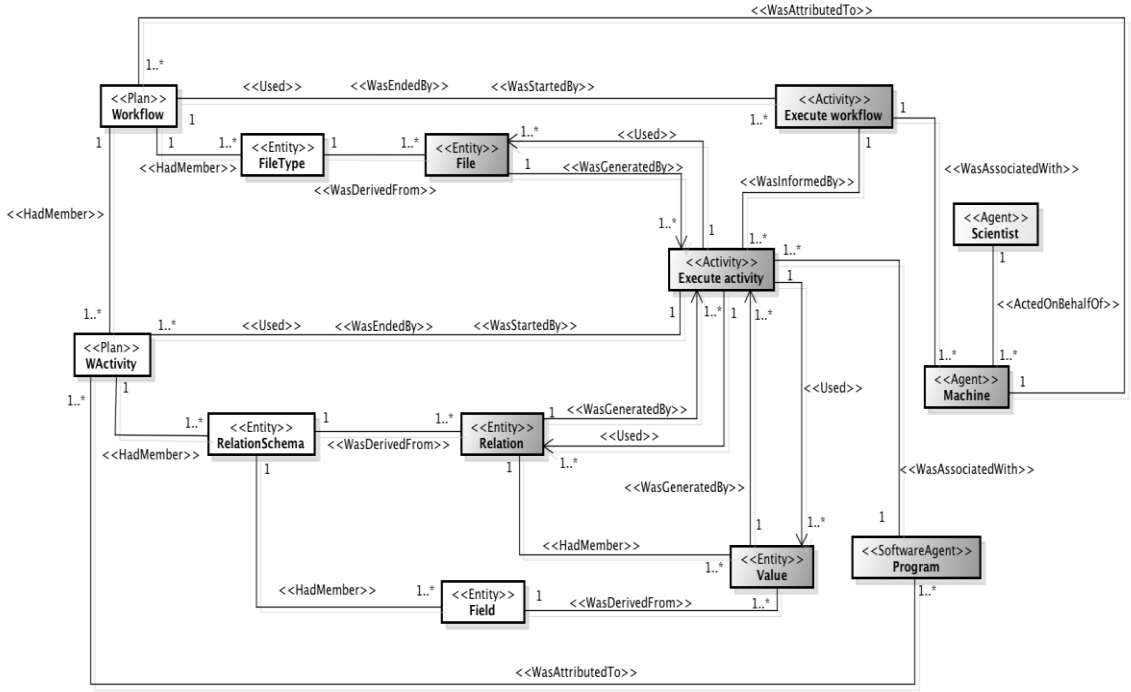

Figura 4 Esquema do Prov-Wf

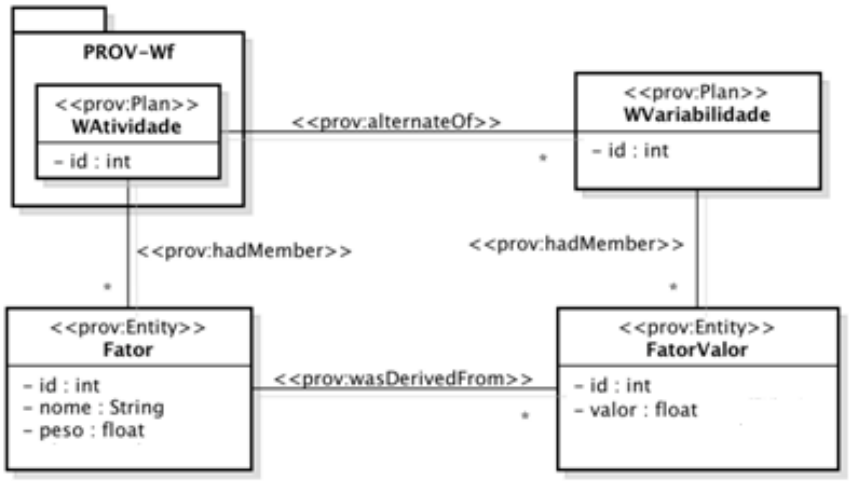

Figura 5 Extensão do PROV-Wf

As alternativas de uma atividade são representadas na entidade *WVariabilidade*, que são as diferentes implementações de um *WAtividade*. A entidade *Fator* representa um aspecto que pode ser levado em consideração durante a execução do *workflow*, como desempenho, qualidade dos dados gerados, etc. Cada fator tem um peso associado. Este peso significa o quanto este fator deve ser favorecido durante a execução do *workflow*. Os pesos podem também ser vistos como uma representação dos objetivos do cientista, traduzindo a maneira como o cientista deseja que o *workflow* seja executado naquele momento. A entidade *FatorValor* representa a relação entre uma *WVariabilidade* e um *Fator*, caracterizando essa relação por meio de um valor. Para cada *Fator* associado a uma *WAtividade*, cada *WVariabilidade* está associada a um *FatorValor* que qualifica a variabilidade quanto ao aspecto do fator em questão. Por exemplo, se para um fator fictício *QualidadeX*, a *WVariabilidade* V_1 tem valor 5, enquanto a *WVariabilidade* V_2 tem valor 8, isso significa que V_2 tem *QualidadeX* melhor que V_1.

3.5 Máquina de workflow

A máquina de *workflow* é o componente responsável por executar as atividades, ou seja, por executar os programas contidos em cada atividade do *workflow* respeitando suas dependências de dados. É preciso que a máquina de *workflow* e o Agente de Adaptação tenham acesso e se guiem pelas mesmas informações de definição do *workflow*. O DynAdapt prevê que estas informações estejam parametrizadas no repositório de proveniência e não estaticamente no plano de execução do *workflow*, de forma que tanto a máquina de execução quanto o agente de adaptação possam interagir com tais informações. Para esta dissertação foi utilizado o Chiron, porém pode-se utilizar qualquer máquina de *workflow* que siga o modelo *PROV-Wf* de proveniência para representar, em tuplas, os passos de execução do *workflow*. O Chiron utiliza a proveniência prospectiva parametrizada no banco de dados para conduzir a execução do *workflow*, portanto as mudanças na proveniência afetam a execução em tempo real. Assim, o Agente de Adaptação implementado nesta dissertação foi construído de modo a manipular o plano de execução presente no repositório de proveniência, que é o plano utilizado pelo Chiron para a execução do workflow. Para que experimentos executados em outras máquinas de workflow possam usufruir dos benefícios do DynAdapt, é necessário o desenvolvimento de um cartucho para o Agente de Adaptação que seja

capaz de manipular o plano de execução utilizado da máquina de workflow correspondente, conforme discutido na Seção 3.3.

4. Avaliação Experimental

Para avaliar o DynAdapt foram executados dois experimentos. O primeiro é um experimento preliminar com um *workflow* sintético, com o objetivo de avaliar o funcionamento da abordagem e apresentar evidências dos possíveis ganhos com o uso do DynAdapt (Santos *et al.* 2013a). O segundo é um experimento com um *workflow* real do domínio da bioinformática, o SciPhy, com o objetivo de avaliar os ganhos que podem ser alcançados com o uso do DynAdapt em um caso real (Santos *et al.* 2013b). Ambos experimentos tratam da troca de atividade em diferentes momentos da execução de seus respectivos *workflows*, mostrando a variação dos ganhos de tempo para os diferentes momentos de troca de atividade. Ao longo deste capítulo são apresentados os detalhes dos experimentos bem como os resultados alcançados.

4.1 Especificações Técnicas de DynAdapt

Para os experimentos apresentados neste trabalho utilizamos a implementação do DynAdapt, desenvolvido em Java, através da ferramenta de desenvolvimento Eclipse versão Helios Service Release 2. Para a máquina de *workflow* utilizamos o Chiron e o SciCumulus, com os seus respectivos repositórios de proveniência armazenados no PostgreSQL. O esquema da base de proveniência utilizado pelo Chiron e pelo SciCumulus foram estendidos para acomodar o modelo de custo ponderado conforme apresentado anteriormente.

4.2 Experimento com Workflow Sintético

Neste primeiro experimento, realizamos uma varredura de parâmetros com um *workflow* sintético W composto pelas três atividades A, B e C e pelas arestas (A,B) e (B,C). Cada instância de cada atividade (*i.e.* ativação) consiste em um programa Java cuja execução dura em média 60 segundos com desvio padrão de 5 segundos. Para simular este tempo de execução utilizamos o método *sleep* da classe *Thread* do Java. Na Figura 6 temos a representação gráfica do *workflow*.

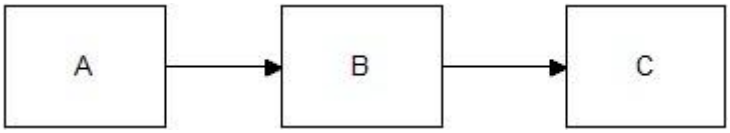

Figura 6 Workflow sintético utilizado no experimento.

À atividade B associamos duas variabilidades, B_1 e B_2. Ou seja, na atividade B, B1 e B2 são compatíveis, possuem os mesmos esquemas de entrada e saída e são alternativas entre si. Ao conjunto de atividades {B, B_1, B_2} foram associados 3 fatores:

custo, confiabilidade, qualidade. Cada fator é qualificado em uma escala de zero a dez conforme a Tabela 2.

Tabela 2 Valores dos fatores de B e suas variabilidades.

	Custo	**Confiabilidade**	**Qualidade**
B	8	6	7
B1	4	9	8
B2	6	5	10

A tabela deve ser interpretada da seguinte forma: executar o *workflow* com B oferece o melhor custo, com B_1 produz a melhor confiabilidade e com B_2 gera resultados com melhor qualidade. Para verificar o funcionamento do DynAdapt simulamos a situação em que o fator mais valorizado é a qualidade, atribuindo a este o peso 0,4, enquanto aos fatores custo e qualidade atribuímos peso 0,3. Ao calcularmos a função de custo para cada alternativa, temos os resultados conforme os apresentados na Tabela 3. Assim, podemos verificar que, considerando as informações das Tabelas 2 e 3, a atividade mais adequada para execução é B_2. Vale lembrar que os ajustes nos pesos e nos fatores podem ser feitos em qualquer momento da execução do *workflow*. Uma vez que estes valores sejam alterados, o DynAdapt recalcula por meio da função de custo ponderado qual a atividade é a mais adequada para execução.

Tabela 3 Cálculo dos custos ponderados de B e suas variabilidades.

	Custo Ponderado
B	8x0,3 + 6x0,3 + 7x0,4 = **7**
B1	4x0,3 + 9x0,3 + 8x0,4 = **7,1**
B2	6x0,3 + 5x0,3 + 10x0,4 = **7,3**

Para avaliar a abordagem proposta, comparamos os tempos de execução do experimento desde o início da execução do *workflow* W até a obtenção do resultado desejado seguindo duas abordagens. Na abordagem manual, o cientista executa o *workflow* uma vez e analisa os resultados. Em seguida, ele realiza a adaptação no

workflow manualmente, que neste caso é a troca de B por B_2, e então executa o *workflow* todo novamente. Na abordagem dinâmica, utilizando o DynAdapt e a proveniência em tempo real, o cientista inicia a execução do *workflow* original W. Uma vez que os resultados parciais sejam suficientes para sua análise (*i.e.* existem dados de proveniência disponíveis), mesmo que a execução não tenha chegado ao fim, o cientista, via DynAdapt, analisa a proveniência e requisita a adaptação da atividade B para B_2. Isto pode ser feito por meio de alterações nos valores dos fatores e pesos ou diretamente por meio de uma requisição explícita de troca da atividade B por B_2.

Para simular o cenário dinâmico, consideramos que o tempo médio de execução do *workflow* é igual a 50 horas e definimos dois cenários de adaptação. Em um cenário, os resultados parciais necessários estão disponíveis após 30% da execução completa do *workflow*. No outro cenário, os resultados parciais estão disponíveis com 60% do *workflow* executado. Desta forma, no primeiro cenário (30%), a alteração dinâmica é feita após cerca de 15 horas enquanto que no segundo cenário (60%), ela é feita após cerca de 30 horas. O DynAdapt realiza a alteração requerida pelo cientista, de acordo com os fatores, de forma que as instâncias em execução sejam imediatamente afetadas pela adaptação. A Figura 7 mostra os resultados obtidos com ambos os cenários.

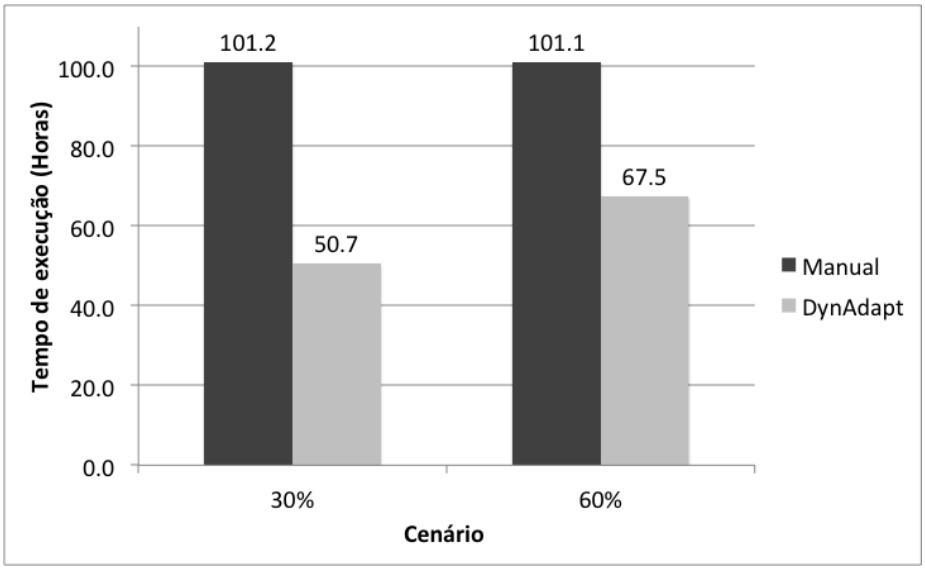

Figura 7 Comparação do tempo total de execução.

Para o cenário de 30%, a atividade B foi trocada por B_2 ainda durante a execução da atividade A, antes de B ser iniciada. Para este caso o uso do DynAdapt gerou uma economia de 49,8% de tempo em relação à abordagem manual. Para o cenário de 60%, a substituição da atividade B por B_2 foi realizada durante a execução de B. Neste caso, as instâncias de B que já estavam em execução foram finalizadas e as que

ainda não tinham sido iniciadas foram abortadas. Em seguida, foram criadas as ativações da atividade B_2. Para este caso, a redução no tempo total de execução do *workflow* foi de 33,2%. Percebemos que a redução do tempo foi maior no cenário 30%. Isso porque no momento da troca, nenhuma instância de B tinha sido iniciada e, portanto, não há a sobrecarga da finalização de instâncias pendentes nem a reexecução de instâncias. Para o cenário 60%, no momento da troca há instancias de B que já estavam em execução. Sendo assim, é necessário esperar a finalização destas instâncias para dar prosseguimento à execução do *workflow* com as instâncias criadas para a atividade B_2.

4.3 Workflow de Análise Filogenética

No segundo experimento desta dissertação avaliamos a troca de atividade em um *workflow* real de análise filogenética, uma área da bioinformática, que foi uma das motivações para este trabalho. A filogenia é um dos campos da bioinformática que visam a comparar centenas de diferentes genomas computacionalmente a fim de identificar semelhanças na evolução de diferentes organismos. Diversos tipos de aplicações computacionais de bioinformática são utilizados na área de filogenia. Como exemplo, podemos citar aplicações de AMS e de Eleição de Modelo Evolutivo, que têm apresentando um aumento em escala e complexidade. O gerenciamento de experimentos de filogenia não é trivial, uma vez que suas atividades fazem uso de computação intensiva e geram uma grande massa de dados. Para apoiar a execução destes experimentos são utilizadas técnicas de *workflows* científicos. Em *workflows* da filogenética cientistas executam um conjunto de atividades a fim de produzir um conjunto de árvores filogenéticas, que são usadas para inferir relações entre as evoluções de genes de diferentes espécies.

O SciPhy (Ocaña *et al.* 2011b) é um exemplo de *workflow* de análise filogenética. O SciPhy é um *workflow* de varredura de parâmetros em que todas as atividades são executadas para cada arquivo dentre o grande conjunto de dados de entrada. Este *workflow* é composto por quatro atividades: (1) Alinhamento Múltiplo de Sequências, (2) Conversão de Alinhamento, (3) Eleição de Modelo Evolutivo e (4) Construção de Árvores Filogenéticas. Para a atividade de Alinhamento Múltiplo de Sequências podem ser utilizadas diferentes implementações como MAFFT, Kalign, ClustalW, Muscle e ProbCons. Para a atividade de Conversão de alinhamento é utilizado o ReadSeq. Para a Eleição do Modelo Evolutivo é utilizada o ModelGenerator.

E para a Construção de Árvores Filogenéticas é utilizado o RAxML. Uma visão geral do SciPhy é apresentada na Figura 8.

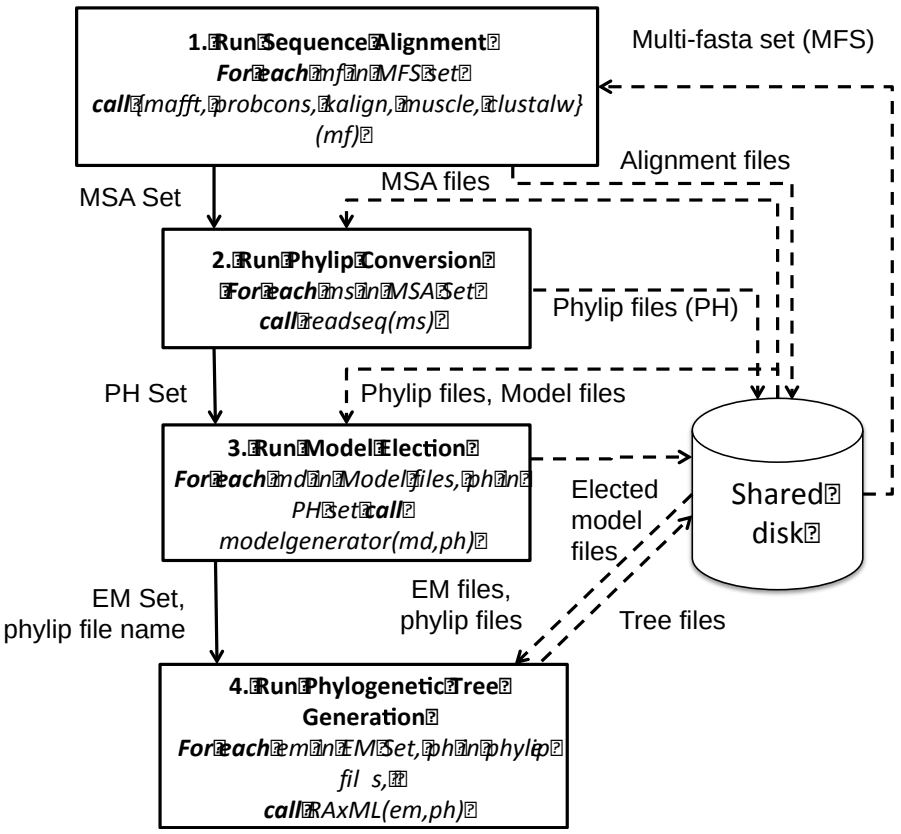

Figura 8 Workflow SciPhy (Ocaña et al. 2011b).

4.4 Experimento com o Workflow SciPhy

Conforme mencionado na seção anterior, a atividade de AMS possui diferentes variabilidades. De acordo com ambiente de execução dentre outros aspectos, uma implementação pode ser mais desejável do que outra. Neste experimento, dada uma execução do SciPhy na nuvem por meio do SciCumulus, simulamos uma alteração nos requisitos do experimento, de forma que uma implementação da atividade de AMS é mais vantajosa do que a implementação que havia sido escolhida para execução a priori. Para este experimento escolhemos os fatores tempo e custo financeiro como base para a tomada de decisão para a alteração no workflow. Podemos escolher outros fatores, como qualidade, por exemplo. Porém, utilizamos apenas os dois fatores mencionados acima, uma vez que queremos testar e avaliar a troca de uma atividade por uma de suas variabilidades. Assim, temos uma variabilidade que tem bom desempenho e outra variabilidade com bom custo financeiro. A ideia é simular uma situação onde, inicialmente, deseja-se minimizar o custo financeiro. E *a posteriori* deseja-se otimizar o

desempenho. Assim, teremos a troca da variabilidade que possui bom custo financeiro pela variabilidade que possui melhor desempenho.

O serviço de computação em nuvem utilizado neste experimento é o Amazon AWS (Amazon EC2 2010). O AWS é um serviço da Amazon que provê diferentes tipos de máquinas virtuais, em que cada máquina virtual tem diferentes configurações de memória e CPU. O tipo de máquina virtual utilizado neste trabalho é m1.large – 7.5GB RAM, 850GB armazenamento com dois núcleos. Cada instância da máquina virtual usa um processado Intel Xeon quad-core de 2.33GHz. O sistema operacional da máquina virtual é o Linux CentOS 5 (64-bit), configurado com as aplicações necessárias para a execução do SciPhy. A imagem virtual (AMI IDs: ami-6ela8907) encontra-se armazenada na nuvem bem como o SciCumulus (Oliveira *et al.* 2010) cria a màquina paralela (*cluster)* virtual baseado nesta imagem.

Para executar o SciPhy exploramos as alternativas da atividade de AMS. O SciPhy utiliza um conjunto de arquivos multi-fasta como entrada. Estes arquivos contêm sequências de proteína. Este conjunto de dados é composto por 200 aminoácidos arquivos multi-fasta e cada arquivo multi-fasta é constituído por uma média de 10 sequências. O MAFFT versão 6.857 foi especificado como a implementação inicial para a atividade de AMS. Os programas alternativos para esta atividade considerados neste experimento foram: ClustalW v.2.1, Kalign v.1.04, Muscle v.3.8.31, e ProbCons v.1.12.

Este experimento visa a troca dinâmica da primeira atividade do SciPhy, o AMS, a fim de alcançar melhor desempenho e menor custo financeiro com a execução na nuvem. À atividade de AMS associamos diferentes fatores como apontados por Oliveira *et al.* (2012b): desempenho, custo financeiro e qualidade. Desta forma, para cada programa alternativo para a atividade de AMS teremos a função de custo calculada como $C = a_1.v_{desempenho} + a_2.v_{custoFinanceiro} + a_3.v_{qualidade}$.

A execução do *workflow* original começa com o uso do MAFFT para a atividade de AMS. Durante a execução do *workflow* o cientista analisa os resultados preliminares e verifica que o desempenho e o custo financeiro não estão de acordo com os requisitos do experimento. Então, o cientista altera os pesos dos fatores indicando que requisitos devem ser mais valorizados nesta execução do *workflow*. Uma vez que os pesos e/ou valores dos fatores foram alterados, o DynAdapt, baseado nos fatores e na função de custo, altera o *workflow* de forma que as instâncias do *workflow* em execução sejam

imediatamente afetadas pela adaptação. Para este experimento, ajustamos os valores dos pesos de modo que o fator tempo é o mais valorizado. Assim, o DynAdapt escolhe a variabilidade que tem melhor desempenho. Desta forma, uma vez que o fator tempo passou a ser o mais valorizado, a ferramenta MAFFT é substituída pelo Muscle. A razão da troca do MAFFT pelo Muscle ocorre porque o MAFFT utiliza um métodos de alinhamento de sequência que tem foco na acurácia ao invés de desempenho. Cada execução do MAFFT é em média 20 vezes mais lenta que a execução do programa Muscle, conforme pode ser visto na Figura 9, onde o eixo horizontal representa o tempo de execução do *workflow* e o eixo vertical representa o tempo médio de execução de cada tarefa que termina em um tempo específico.

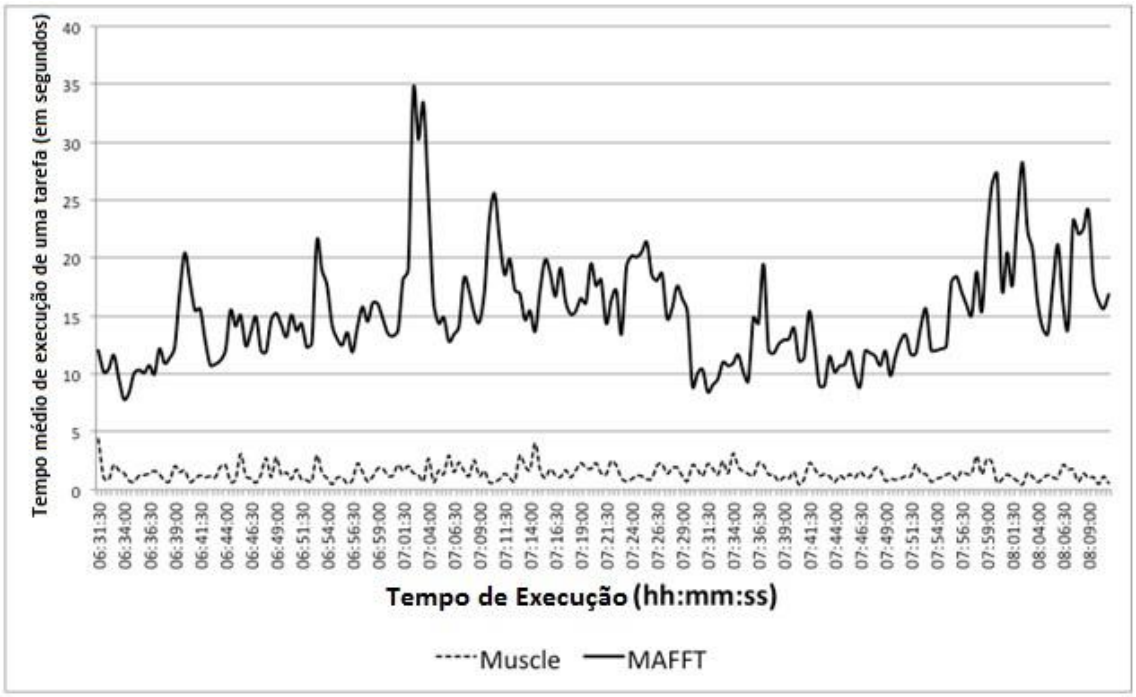

Figura 9 Média de tempo de execução Muscle versus MAFFT.

A fim de avaliar a abordagem proposta nesta dissertação e analisar a diferença entre o tempo total de execução e custo financeiro, analisamos três cenários: (i) troca do MAFFT pelo Muscle quando 10% dos arquivos de entrada foram processados, (ii) troca do MAFFT pelo Muscle quando 50% dos arquivos de entrada foram processados e (iii) troca do MAFFT pelo Muscle quando 90% dos arquivos de entrada foram processados. Para cada cenário comparamos o tempo total de execução do experimento e o custo financeiro.

Na Figura 10 temos uma visão geral dos resultados das execuções segundo os três cenários propostos. O gráfico deixa claro que o MAFFT é substituído pelo Muscle

nos pontos onde o tempo médio de execução de uma tarefa apresenta queda abrupta. A alteração dinâmica impacta diretamente no desempenho do *workflow* como um todo. Podemos notar uma diferença de aproximadamente 40 minutos entre o tempo total do *workflow* com a troca do MAFFT com 10% e com 90% dos dados processados.

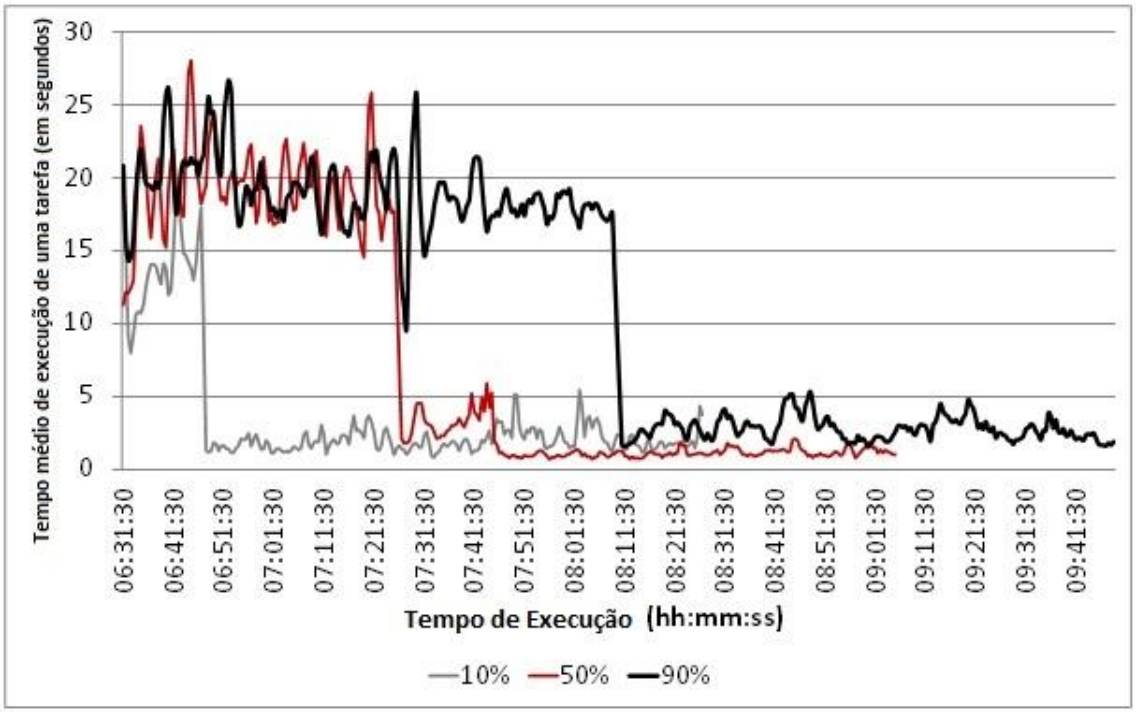

Figura 10 Alterações dinâmicas com o SciPhy.

Apesar de o SciPhy ser composto por várias atividades, nesta análise focamos no tempo total de execução e no custo financeiro associados à atividade de Alinhamento Múltiplo de Sequências. A execução do *workflow* utilizando somente a ferramenta MAFFT, para 200 arquivos multi-fasta de entrada, leva aproximadamente 20 horas. Já a execução do *workflow* utilizando somente a ferramenta Muscle, com os mesmos 200 arquivos multi-fasta de entrada, leva aproximadamente 8 horas. A Figura 11 apresenta o tempo total de execução com a utilização do DynAdapt (barra cinza) e com a Re-execução do *workflow* (barra pontilhada). Em geral temos uma diferença de 14% entre o tempo total com o uso do DynAdapt e com o uso da re-execução manual do *workflow*. Este resultado pode significar um considerável valor absoluto de tempo uma vez que os experimentos crescem em complexidade e volume de dados processados. Este ganho de desempenho é esperado uma vez que o DynAdapt não requer que o *workflow* seja totalmente reexecutado, economizando tempo de instalação e inicialização, principalmente em ambientes como a nuvem. Com a utilização do DynAdapt, o tempo total de execução foi reduzido de 20 horas, que é o tempo médio gasto com a execução

do *workflow* somente com o MAFFT para 8 horas e 6 minutos, que foi o tempo total de execução do *workflow* com a troca dinâmica ocorrendo após 10% dos dados processados. Esta redução representa 40% de redução do tempo total de execução do *workflow*. Quanto ao custo financeiro, uma vez que o tempo total de execução do *workflow* sobre o ambiente de nuvem da Amazon foi reduzido, o custo financeiro associado também diminuiu, conforme pode ser visto na Figura 11.

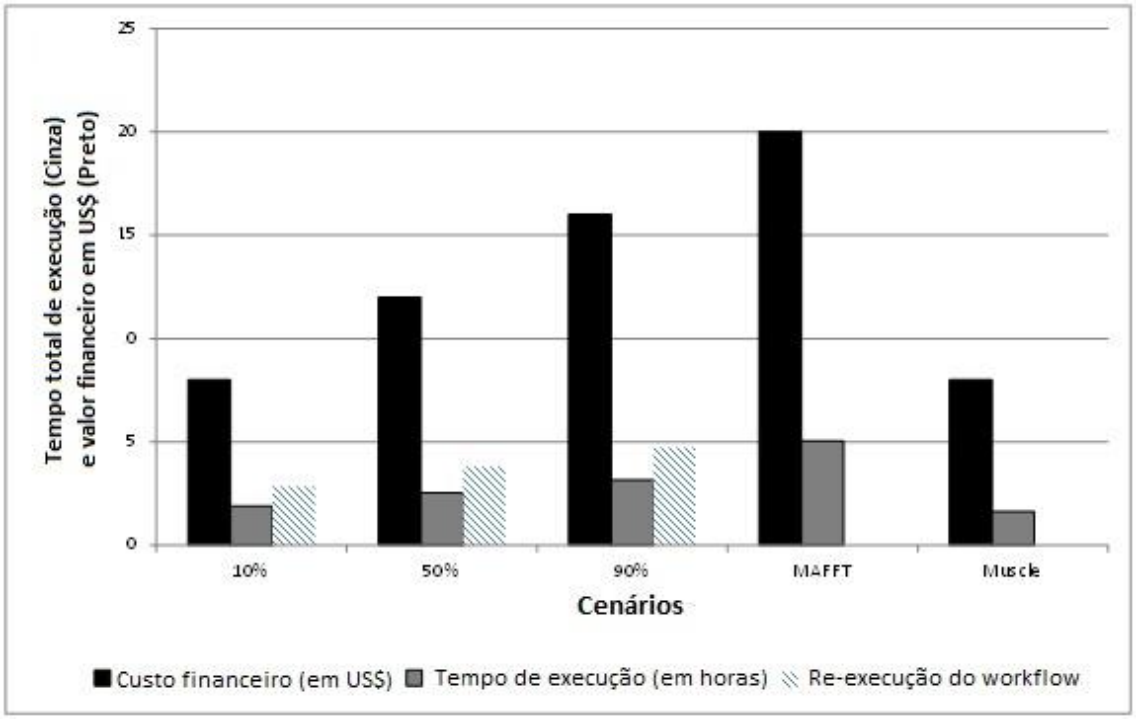

Figura 11 Comparação de tempo de execução e custo financeiro.

5. Conclusões

A natureza exploratória dos experimentos científicos torna os *workflows* científicos objetos de constantes alterações dinâmicas. Na fase de análise dos resultados do experimento, o cientista toma decisões sobre as alterações a serem feitas no *workflow* para que numa execução posterior os resultados sejam mais próximos do desejado. Em geral, caso o cientista necessite alterar a especificação do *workflow* em tempo de execução, ele precisa abortar a execução corrente, fazer a alteração e, então reexecutar o *workflow*. Isto ocorre porque, em geral, os resultados e os dados de proveniência só estão disponíveis ao fim da execução do *workflow* e os SGWfC existentes não permitem alterações no *workflow* em execução. Em muitos casos os resultados parciais da execução já são suficientes para a tomada de decisão sobre as alterações a serem feitas no experimento. A tarefa de reexecutar todo o experimento é complexa e tende a utilizar mais recursos. Para o caso de *workflows* executados na nuvem, por exemplo, o volume de dados trafegados para a reinstalação do *workflow* aumenta na medida em que, a cada alteração, o *workflow* precisa ser reinstalado para, então, ser reexecutado. Além disso, as atividades que não foram alteradas precisam ser reexecutadas todas novamente.

Uma atividade de um *workflow* científico pode possuir um conjunto de diferentes implementações que podem ser vistas como alternativas entre si, uma vez que executem a mesma tarefa. De acordo com o ambiente e com os requisitos do experimento uma variabilidade pode se tornar mais vantajosa do que outra. Nos trabalhos existentes, a troca de atividades só pode ser realizada após a interrupção da execução, a redefinição da especificação do workflow com a nova atividade e re-submissão de sua execução. Apresentamos um formalismo que permite a definição de atividades alternativas junto ao grafo que especifica o workflow. A fim de permitir trocas dinâmicas das atividades de um *workflow* conduzidas pelo cientista, desenvolvemos o DynAdapt. Com o DynAdapt o cientista deve pré-configurar as ferramentas alternativas para cada atividade em um conjunto de variabilidades. A cada conjunto de variabilidades associamos um conjunto de fatores, que representa os aspectos relevantes para a execução do experimento. A cada fator, o cientista associa um peso que representa o quanto tal fator é valorizado na execução do experimento. Estes pesos e fatores podem ser alterados a qualquer momento pelo cientista, mesmo durante a execução do *workflow*. Por meio de um modelo de custo ponderado, o DynAdapt elege a variabilidade que dever ser utilizada e realiza a troca da atividade

pela variabilidade escolhida. Esta troca é realizada diretamente no plano de execução da máquina de *workflow*, de forma que as instâncias em execução são imediatamente afetadas. Além de trocar a atividade por meio de do modelo de custo, o DynAdapt permite que o cientista realize a troca de uma atividade por uma variabilidade arbitrariamente.

Avaliamos o DynAdapt por meio de dois experimentos. Em ambos os casos, o ganho obtido com a abordagem proposta neste trabalho foi medido por meio da análise do tempo gasto desde o início da execução do *workflow* até a obtenção dos resultados desejados. No primeiro experimento, avaliamos a troca de atividade com 30% e com 60% dos dados de entrada consumidos. Para o caso da troca com 30% dos dados consumidos o uso do DynAdapt gerou uma economia de quase 50% no tempo total de execução do experimento, comparado ao tempo total com a abordagem de reexecução manual do *workflow*. Este resultado preliminar serviu como uma primeira avaliação para a abordagem e mostrou que seria possível alcançar ganhos de tempo significativos com a utilização do DynAdapt. Já no segundo experimento, avaliamos o uso do DynAdapt com um *workflow* real, o SciPhy, um *workflow* de análise filogenética. Neste segundo experimento, avaliamos a execução do SciPhy com o DynAdapt no ambiente de nuvem da Amazon, o AWS. O *workflow* SciPhy pode utilizar diferentes implementações para a atividade de AMS, sua primeira atividade. Neste experimento, o SciPhy foi configurado para iniciar a execução com o MAFFT e para trocar a primeira atividade após 10%, 50% e 90% dos dados de entrada consumidos. À primeira atividade do SciPhy atribuímos maiores pesos aos fatores desempenho e custo financeiro, uma vez que são aspectos relevantes para a execução de *workflow* na nuvem. O DynAdapt elegeu o Muscle como a melhor variabilidade para a atividade de alinhamento de sequências múltiplas, de acordo com o critério que definimos no experimento. Este experimento mostrou que a utilização do DynAdapt gerou economia de 40% no tempo de execução, comparado ao tempo gasto com a reexecução manual do *workflow*. Consequentemente, o DynAdapt gerou também uma economia nos custos financeiros gastos com a execução na nuvem.

Os experimentos mostraram também que o DynAdapt permite ao cientista conduzir dinamicamente a execução do experimento, seja por meio dos fatores, seja por meio da troca arbitrária de uma atividade por uma de suas variabilidades.

Como o DynAdapt permite a condução do experimento pelo cientista, seria interessante um interface gráfica para facilitar a comunicação entre o usuário e o DynAdapt. Uma ideia para esta interface seria uma espécie de painel de controle, onde o cientista, por meio de recursos gráficos, possa manipular os pesos e fatores associados ao seu experimento, ou escolher uma variabilidade para uma troca arbitrária. Informações como percentual de dados consumidos, tempo dispensado, custo financeiro associado, etc poderiam também ser apresentadas nesta interface a fim de facilitar a decisão sobre uma troca dinâmica de atividade.

Outro desdobramento interessante é o de preenchimento dos valores dos fatores associados a um conjunto de variabilidades. Atribuir uma "nota" a uma atividade segundo um fator não é uma tarefa trivial. Para o caso de fatores quantitativos, como o fator tempo, por exemplo, poderíamos automatizar este preenchimento a partir dos dados de proveniência referentes a tempo de execução da atividade. Desta forma, poderíamos realimentar o modelo de fatores periodicamente sem a necessidade de o cientista ter que alterar cada fator manualmente a cada alteração de seu valor. Porém, para fatores qualitativos, como qualidade dos dados gerados, o preenchimento automático de valores deste fator se torna mais complexo, uma vez que é preciso uma análise dos resultados gerados e um conjunto de critérios para avaliar a qualidade.

Referências Bibliográficas

Abouelhoda, M., Issa, S., Ghanem, M., (2012), "Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support", *BMC Bioinformatics*, v. 13, p. 77.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., Mock, S., (2004), "Kepler: an extensible system for design and execution of scientific workflows". In: *Scientific and Statistical Database Management*, p. 423–424, Greece.

Amazon EC2, (2010), *Amazon Elastic Compute Cloud (Amazon EC2)*, http://aws.amazon.com/ec2/.

Beveridge, W. I. B., (2004), *The Art of Scientific Investigation*. Blackburn Press.

Boeres, C., Sardiña, I., Drummond, L., (2011), "An efficient weighted bi-objective scheduling algorithm for heterogeneous systems", *Parallel Computing*, v. 37, n. 8 (Agosto.), p. 349–364.

Bowers, S., Ludascher, B., Ngu, A. H. H., Critchlow, T., (2006), "Enabling ScientificWorkflow Reuse through Structured Composition of Dataflow and Control-Flow". In: *Proceedings of the 22nd International Conference on Data Engineering Workshops*, p. 70

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.

Costa, F., Oliveira, D., Ocaña, K., Ogasawara, E., Dias, J., Mattoso M., (2012a), "Handling Failures in Parallel Scientific Workflows Using Clouds". , Salt Lake City.

Costa, F., Oliveira, D., Ocaña, K., Ogasawara, E., Mattoso, M., (2012b), "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: *Proc. of the International Workshop on Managing and Querying Provenance Data at Scale*, Genova, Italy.

Culler, D. E., Singh, J. P., Gupta, A., (1999), *Parallel computer architecture: a hardware/software approach*. 2 ed.

Dantas, M., (2005), "Clusters Computacionais", *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais*, 1 edRio de Janeiro: Axcel Books, p. 145–180.

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.

Deelman, E., Mehta, G., Singh, G., Su, M.-H., Vahi, K., (2007), "Pegasus: Mapping Large-Scale Workflows to Distributed Resources", *Workflows for e-Science*, Springer, p. 376–394.

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 Workshop on Workflows in Support of Large-Scale Science*, p. 31–36, Seattle, WA, USA.

Eddy, S. R., (1996), "Hidden Markov models", *Current Opinion in Structural Biology*, v. 6, n. 3 (Jun.), p. 361–365.

Eisen, J. A., (2003), "Phylogenomics: Intersection of Evolution and Genomics", *Science*, v. 300, n. 5626 (Jun.), p. 1706–1707.

Ellis, C., Keddara, K., Rozenberg, G., (1995), "Dynamic change within workflow systems". , p. 10 – 21, New York, NY, USA.

Emeakaroha, V. C., \Labaj, P., Maurer, M., Brandic, I., Kreil, D. P., (2011), "Optimizing bioinformatics workflows for data analysis using cloud management techniques". In: *Proceedings of the 6th workshop on Workflows in support of large-scale science*, p. 37–46, New York, NY, USA.

Fabian, N., Moreland, K., Thompson, D., Bauer, A. C., Marion, P., Geveci, B., Rasquin, M., Jansen, K. E., (Oct.), "The ParaView Coprocessing Library: A scalable, general purpose in situ visualization library". In: *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, p. 89–96

Foster, I., Kesselman, C., (2004), *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

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.

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

Hoheisel, A., Alt, M., (2007), "Petri Nets", *Workflows for e-Science*, Springer, p. 190–207.

Horta, F., Dias, J., Elias, R., Oliveira, D., Coutinho, A. L. G. A., Mattoso, M., (2013), "Prov-Vis: Large-Scale Scientific Data Visualization Using Provenance (Abstract)". In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, Denver, CO, USA.

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

Jr, H. G. G., (2002), *Scientific Method in Practice*. 1st ed. Cambridge University Press.

Kammer, P. J., Bolcer, G. A., Taylor, R. N., Hitomi, A. S., Bergman, M., (2000), "Techniques for Supporting Dynamic and Adaptive Workflow", *Computer Supported Cooperative Work (CSCW)*, v. 9, n. 3-4, p. 269–292.

Lee, K., Sakellariou, R., Paton, N. W., Fernandes, A., (2007), "Workflow adaptation as an autonomic computing problem". In: *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, p. 29–34, New York, NY, USA.

Mattoso, A., Silva, F., Ruberg, N., Cruz, M., (2008), "Gerência de Workflows Científicos: Uma Análise Crítica no Contexto da Bioinformática", *COPPE/UFRJ*, n. Relatório técnico

Mattoso, M., Dias, J., Oliveira, D., Ocaña, K., Ogasawara, E., Costa, F., Horta, F., Silva, V., Araújo, I., (2013), "User-Steering of HPC Workflows: State of the Art and Future Directions". In: *Proceeding of the 2nd International Workshop on Scalable Workflow Enactment Engines and Technologies (SWEET'13)*, New York, NY, 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.

Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T., Goble, C., (2010), "Taverna, reloaded". In: *Proceedings of the 22nd international conference on Scientific and statistical database management*, p. 471–481, Berlin, Heidelberg.

Moreau, L., Missier, P., Belhajjame, K., Cresswell, S., Golden, R., Groth, P., Miles, S., Sahoo, S., (2011). The PROV Data Model and Abstract Syntax Notation. Disponível em: http://www.w3.org/TR/prov-dm/. Acesso em: 14 Dec 2011.

Ocaña, K. A. C. S., Oliveira, D. de, Horta, F., Dias, J., Ogasawara, E., Mattoso, M., (2012), "Exploring Molecular Evolution Reconstruction Using a Parallel Cloud-based Scientific Workflow", *Advances in Bioinformatics and Computational Biology*, , chapter 7409, Berlin, Heidelberg: Springer, p. 179–191.

Ocaña, K. A. C. S., Oliveira, D., Dias, J., Ogasawara, E., Mattoso, M., (2011a), "Optimizing Phylogenetic Analysis Using SciHmm Cloud-based Scientific Workflow". In: *2011 IEEE Seventh International Conference on e-Science (e-Science)*, p. 190–197, Stockholm, Sweden.

Ocaña, K. A. C. S., Oliveira, D., Ogasawara, E., Dávila, A. M. R., Lima, A. A. B., Mattoso, M., (2011b), "SciPhy: A Cloud-Based Workflow for Phylogenetic Analysis of Drug Targets in Protozoan Genomes", In: Norberto de Souza, O., Telles, G. P., Palakal, M. [eds.] (eds), *Advances in Bioinformatics and Computational Biology*, , chapter 6832, Berlin, Heidelberg: Springer, p. 66–70.

Ogasawara, E., Dias, J., Oliveira, D., Porto, F., Valduriez, P., Mattoso, M., (2011), "An Algebraic Approach for Data-Centric Scientific Workflows", *Proc. of VLDB Endowment*, v. 4, n. 12, p. 1328–1339.

Ogasawara, E., Dias, J., Silva, V., Chirigati, F., Oliveira, D., Porto, F., Valduriez, P., Mattoso, M., (2013), "Chiron: A Parallel Engine for Algebraic Scientific Workflows", *Concurrency and Computation*, v. 25, n. 16, p. 2327–2341.

Ogasawara, E., Paulino, C., Murta, L., Werner, C., Mattoso, M., (2009), "Experiment Line: Software Reuse in Scientific Workflows". In: *Scientific and Statistical Database Management*, p. 264–272, New Orleans, Louisiana, USA.

Oliveira, D., Ocaña, K. A. C. S., Ogasawara, E., Dias, J., Goncalves, J., Mattoso, M., (2012a), "Cloud-based Phylogenomic Inference of Evolutionary Relationships: A Performance Study". In: *Proceedings of the 2nd International Workshop on Cloud Computing and Scientific Applications (CCSA)*, Ottawa, Canadá.

Oliveira, D., Ocaña, K., Baião, F., Mattoso, M., (2012b), "A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds", *Journal of Grid Computing*, v. 10, n. 3, p. 521–552.

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.

Pintas, J., Oliveira, D., Ocaña, K. A. C. S., Ogasawara, E., Mattoso, M., (2013), "SciLightning: a Cloud Provenance-based Event Notification for Parallel Workflows". In: *Proceedings of the 3rd International Workshop on Cloud Computing and Scientific Applications (CCSA)*, Berlin, Germany.

Pohl, K., Böckle, G., Linden, F. J. van der, (2005), *Software Product Line Engineering: Foundations, Principles and Techniques*. 2005 ed. Springer.

Samak, T., Gunter, D., Goode, M., Deelman, E., Mehta, G., Silva, F., Vahi, K., (2011), "Failure prediction and localization in large scientific workflows". In: *Proceedings of the 6th workshop on Workflows in support of large-scale science*, p. 107–116, New York, NY, USA.

Santos, I., Dias, J., Oliveira, D., Ogasawara, E., Mattoso, M., (2013a), "DynAdapt: Alterações na Definição de Atividades de Workflows Científicos em Tempo de Execução". In: *VII e-Science*, p. 1–8, Maceio, Alagoas, Brazil.

Santos, I., Dias, J., Oliveira, D., Ogasawara, E., Ocaña, K., Mattoso, M., (2013b), "Runtime Dynamic Structural Changes of Scientific Workflows in Clouds". In: *Proceedings of the International Workshop on Clouds and (eScience) Applications Management - CloudAM*, Dresden, Germany.

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

Taylor, I., Shields, M., Wang, I., Harrison, A., (2007b), "The Triana Workflow Environment: Architecture and Applications", *Workflows for e-Science*, Springer, p. 320–339.

Travassos, G. H., Barros, M. O., (2003), "Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering". In: *2nd Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*, p. 117–130, Rome, Italy.

Truong, H., Brunner, P., Fahringer, T., Nerieri, F., Samborski, R., Balis, B., Bubak, M., Rozkwitalski, K., (Dec.), "K-WfGrid Distributed Monitoring and Performance Analysis Services for Workflows in the Grid". In: *Second IEEE International Conference on e-Science and Grid Computing, 2006. e-Science '06*, p. 15–15

Vahi, K., Harvey, I., Samak, T., Gunter, D., Evans, K., Rogers, D., Taylor, I., Goode, M., Silva, F., Al-Shakarchi, E., Mehta, G., Jones, A., Deelman, E., (2012), "A General Approach to Real-time Workflow Monitoring".

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.

Wilson Jr, E. B., (1991), *An Introduction to Scientific Research*. Rev Sub ed. Dover Publications.

Wozniak, J. M., Armstrong, T. G., Maheshwari, K., Lusk, E. L., Katz, D. S., Wilde, M., Foster, I. T., (2012), "Turbine: a distributed-memory dataflow engine for extreme-scale many-task applications". , p. 1–12

Zhao, Y., Hategan, M., Clifford, B., Foster, I., von Laszewski, G., Nefedova, V., Raicu, I., Stef-Praun, T., Wilde, M., (2007), "Swift: Fast, Reliable, Loosely Coupled Parallel Computation". In: *3rd IEEE World Congress on Services*, p. 206, 199, Salt Lake City, USA.