

DynAdapt: Alterações na Definição de Atividades de Workflows Científicos em Tempo de Execução

Igor de Araújo dos Santos¹, Jonas Dias¹, Daniel de Oliveira²,
Eduardo Ogasawara^{1,3}, Marta Mattoso¹

¹COPPE/Universidade Federal do Rio de Janeiro (UFRJ), Brasil

²Universidade Federal Fluminense (UFF), Brasil

³CEFET/RJ, Brasil

{igor, jonasdias, marta}@cos.ufrj.br, danielcmo@ic.uff.br,
eogasawara@cefet-rj.br

Abstract. *Scientific workflows can represent experiments based on computer simulations. Generally, workflows executed in parallel are time consuming and manage a large amount of data. Such characteristics may make the exploratory process of the experiment more difficult or very expensive. In this scenario, it is necessary to handle workflows with dynamic aspects, which allow for changes in workflow definitions during runtime. This way, this article proposes an approach to allow for changes in the definition of the activities of the workflow during runtime, according to criteria defined by scientists.*

Resumo. *Os workflows científicos são capazes de representar experimentos baseados em simulações computacionais. Em geral, workflows executados em paralelo manipulam uma grande massa de dados e demandam um elevado tempo de execução. Tais características podem dificultar ou encarecer o processo exploratório de um experimento. 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. Assim, a proposta deste artigo é oferecer uma abordagem para permitir alterações na definição de atividades de workflows em tempo de execução, de acordo com critérios definidos pelos cientistas.*

1. Introdução

Os *workflows* científicos vêm sendo usados como uma abstração para representar experimentos científicos (Deelman et al. 2009). Um *workflow* científico é composto por atividades, que representam as etapas do experimento, e ligações entre essas atividades, que definem as dependências de dados entre as mesmas. Diante da crescente complexidade das análises dos experimentos e do desenvolvimento da ciência em larga escala, muitos experimentos científicos são executados em grades, *clusters* e mais recentemente em nuvens de computadores, lançando novos desafios relacionados ao controle e monitoramento da execução dos *workflows* nestes ambientes distribuídos. Para a composição, execução e análise dos *workflows* científicos, cientistas utilizam ferramentas como os Sistemas de Gerência de *Workflows* Científicos (SGWfC).

Os *workflows* podem passar por uma série de alterações na sua definição e nos seus parâmetros para que se atinja a configuração que produza o resultado desejado para o experimento. Parte destas alterações é realizada 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 processado em seguida. Ou seja, de acordo com algum resultado (seja ele intermediário ou final), o cientista decide o que deverá ser ajustado na execução do

workflow científico. Parâmetros e dados de entrada são alterados até que o objetivo da execução do *workflow* seja alcançado. Os ajustes mais comuns são relativos aos dados de entrada e os parâmetros. Os cientistas comumente exploram diferentes conjuntos de dados e parâmetros e executam o *workflow* repetidas vezes para descobrir qual a melhor configuração. Entretanto, não são apenas os parâmetros e os dados que podem ser ajustados e variados. Muitas vezes os cientistas, baseados nos resultados de execuções anteriores, realizam ajustes na própria definição do *workflow*, alterando os programas associados às suas atividades a fim de obter um desempenho melhor, um resultado com mais qualidade ou mesmo um programa mais adequado ao perfil dos dados que estão sendo gerados.

Pode ocorrer o caso em que uma mesma atividade do *workflow* tenha várias implementações (programas) correspondentes. Na Bioinformática, por exemplo, uma atividade de alinhamento de sequências pode utilizar diferentes ferramentas como o MAFFT, ProbCons, Muscle, ClustalW ou Kalign (Ocaña *et al.* 2011). Assim, estes programas podem ser vistos como alternativos entre si e, 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, por exemplo, 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 diversos aspectos que podem influenciar a execução do *workflow*. Cada alternativa para uma determinada atividade pode ter um comportamento diferente de acordo com estes aspectos. Sem contar que estes aspectos podem variar durante a execução do *workflow*. Deste cenário, surge a necessidade de um apoio dinâmico para a escolha de uma atividade alternativa. Ou seja, a possibilidade de alterar as atividades de um *workflow* em execução sem que se faça necessário interromper a execução e re-executá-lo por completo para que as alterações sejam consideradas. A necessidade de se trabalhar com *workflows* dinâmicos foi levantada em (Gil *et al.* 2007) como um desafio, porém até hoje esse problema se mantém aberto. Com as soluções existentes hoje, os cientistas precisam interromper ou esperar o término da execução para realizar ajustes no *workflow* para então re-executá-lo por completo.

Cada execução do *workflow* pode demandar um tempo elevado de execução. Atrelado ao tempo de execução pode existir ainda 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 pode trazer economia permitindo que os cientistas realizem ajustes no *workflow* e poupem a execução de parte do *workflow* novamente. Um dos objetivos deste trabalho é permitir que os ajustes realizados se reflitam em tempo real sobre os *workflows* em execução no momento, com o intuito de apoiar o cientista a obter os resultados finais do experimento mais rapidamente e com maior qualidade.

Este artigo apresenta uma abordagem de apoio à adaptação de atividades de *workflows* científicos em tempo de execução. Esse apoio ao cientista visa à realização de mudanças estruturais dinâmicas nas atividades do *workflow*, ou seja, durante a sua execução. Este objetivo é apoiado pela hipótese de que, se permitirmos alterações dinâmicas, o tempo total de execução gasto para a obtenção dos resultados experimentais desejados pode ser minimizado conforme apontado por Dias *et al.* (2011). Conseqüentemente, os custos atrelados à execução do experimento também são reduzidos. Na abordagem proposta, esse dinamismo se dá por meio da troca de uma atividade por outra equivalente durante a execução do *workflow*. Essas trocas seguem critérios pré-estabelecidos pelo cientista, seja ele de qualidade ou de desempenho, por exemplo. Para possibilitar tal adaptação, em tempo de execução, foi desenvolvido um protótipo junto a máquina de execução de *workflows* científicos Chiron (Ogasawara *et al.* 2013).

Este texto está organizado da seguinte forma: A Seção 2 discute os trabalhos relacionados. A Seção 3 descreve a arquitetura do DynAdapt e explica cada um dos componentes. A Seção 4 apresenta uma avaliação experimental e a Seção 5 conclui.

2. Trabalhos relacionados

Na literatura podem-se encontrar alguns trabalhos que evidenciam a natureza dinâmica dos *workflows* científicos e a importância das informações de dados de proveniência para a decisão sobre as alterações dinâmicas que podem ser realizadas. Em Gil *et al.* (2007) foi realizada uma análise sobre os desafios a serem alcançados no contexto dos *workflows* científicos onde é apresentada uma série de requisitos que devem ser atendidos pelos SGWfC, com o avanço da ciência em larga escala. Dentre os requisitos enumerados no texto, destacam-se os requisitos relacionados às características dinâmicas dos *workflows* científicos. Segundo os autores é necessário que os SGWfC deem apoio às alterações dinâmicas que ocorrem nos experimentos, sejam elas alterações semânticas, de ambiente, de estrutura ou por outras necessidades. Entretanto, poucas são as contribuições nesse sentido.

Em Ocaña *et al.* (2011) é proposto um *workflow* chamado SciHmm para, utilizando cadeias de Markov escondidas (HMM), escolher o algoritmo de alinhamento genético mais adequado de acordo com as informações de proveniência de execuções anteriores. Diferentemente da abordagem proposta neste trabalho, o artigo não trata do problema de alteração dinâmica, mas evidencia a possibilidade e a importância de, a partir dos dados de proveniência, uma determinada implementação de uma atividade ser mais adequada do que outras. Dias *et al.* (2011) propõem uma primeira abordagem para apoiar o ajuste dinâmico do *workflow*. Diferentemente da abordagem DynAdapt, o trabalho de Dias *et al.* não tem o foco nas mudanças das atividades, mas sim da exploração de parâmetros do *workflow* por meio da utilização de estruturas de controle que permitem que sejam tomadas decisões sobre o ajuste dos parâmetros, de acordo com a proveniência e os resultados parciais obtidos em tempo de execução. Esta estrutura de controle pode ser programada para desencadear o ajuste automaticamente de acordo com informações de proveniência ou este pode ser feito manualmente pelo cientista. Na literatura pode-se encontrar também alguns trabalhos que tratam das alterações dinâmicas na estrutura do *workflow* como em Ellis *et al.* (1995), Kammer *et al.*, e Tudruj *et al.* (2007), por exemplo. Porém, estes trabalhos são voltados para *workflows* de negócio e para o trabalho cooperativo. Dentro do que foi estudado para a construção do DynAdapt, não foi encontrado SGWfC que permite a alteração dinâmica em atividades de *workflows* científicos.



Figura 1. Visão geral da arquitetura proposta.

3. A Abordagem DynAdapt

Apresentamos uma arquitetura baseada em componentes para apoiar à alteração dinâmica das atividades de um *workflow* de forma automática e baseada em dados de proveniência providos em tempo de execução (Costa *et al.* 2013). A ideia básica é que, dados os aspectos

mais importantes para o cientista quanto à execução do *workflow*, a abordagem, baseada nas informações de proveniência, escolha qual a alternativa de uma atividade é a mais adequada para ser executada no momento. A Figura 1 mostra uma visão geral da arquitetura proposta. A seguir apresentamos cada componente da arquitetura com mais detalhes.

3.1. O Repositório de Proveniência

A proveniência é toda a informação da história do *workflow*, o seu *pedigree* (Freire *et al.* 2008). Ela é fundamental para a validação e reprodutibilidade dos resultados de um experimento. O repositório de proveniência figura como o componente central da arquitetura proposta. Todos os outros componentes interagem diretamente com o repositório, que pode ser consultado durante a execução do *workflow*. Neste repositório são armazenadas todas as informações referentes à definição e à execução do *workflow*, ou seja, proveniência prospectiva e retrospectiva.

O modelo de proveniência adotado é uma extensão do *PROV-Wf* (Costa *et al.* 2013) que é um modelo projetado para representar dados de proveniência de diferentes SGWfC em um único modelo e permitir consultas durante a execução do *workflow*. Ele é baseado no padrão 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 sobre o *PROV-Wf* projetada para este trabalho consiste na inclusão da entidade que representa uma alternativa de uma atividade do *workflow*, chamada variabilidade, inspirada nos conceitos de linha de experimento (Ogasawara *et al.* 2009), e na inclusão do modelo ponderado de fatores, descrito na Seção 4.1.2. Como o *PROV-Wf* é guiado pela abordagem algébrica para *workflows* científicos, toda atividade Y_i do *workflow* consome um conjunto $\{R_{i1}, \dots, R_{in}\}$ de relações com esquema $\{\mathcal{R}_{i1}, \dots, \mathcal{R}_{in}\}$ e produz uma relação de saída T com esquema \mathcal{F} . No modelo proposto, cada atividade pode estar associada a um conjunto de variabilidades, ou seja $Y_i \in \{V_1, \dots, V_k\}$. Logo, para cada relação $R_{ij}(\mathcal{R}_{ij})$ de entrada de Y_i , $\forall V_x \in \{V_1, \dots, V_k\}$ possui uma relação entrada $R_{iv}(\mathcal{R}_{iv})$ tal que $\mathcal{R}_{iv} \subseteq \mathcal{R}_{ij}$. Ou seja, toda a variabilidade de uma atividade Y_i precisa ser capaz de consumir o mesmo conjunto de relações de entrada de Y_i . Toda a informação necessária para utilizar as rotinas de adaptação implementadas neste trabalho são obtidas seguindo o modelo *PROV-Wf*. Assim, qualquer sistema que utilize este mesmo modelo pode se beneficiar das operações construídas para esta abordagem. A Figura 2 apresenta a extensão feita ao *PROV-Wf* mostrando apenas os atributos principais das entidades. Os <<estereótipos>> indicam a representação de cada elemento na ontologia do PROV.

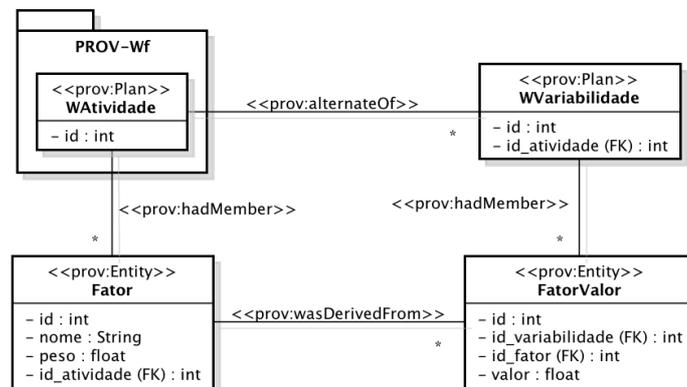


Figura 2. Modelo ponderado de fatores

A entidade *WVariabilidade* representa as alternativas de uma atividade, ou seja, as diferentes implementações de uma *WAtividade*. A entidade *Fator* representa um aspecto que deve ser levado em consideração durante a execução do *workflow*, por exemplo: desempenho, qualidade dos dados gerados, etc. Um fator tem um peso associado, que

significa o quanto este fator deve ser favorecido durante a execução do *workflow*. A entidade *FatorValor* representa a relação entre uma *WVariabilidade* e um *Fator*, caracterizando essa relação através de um valor. Para cada *Fator* associado a uma *WAtividade*, cada *WVariabilidade* está associada a um *FatorValor* que qualifica a variabilidade sob o 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_1 tem *QualidadeX* melhor que V_2 .

3.1.2. O Modelo Ponderado de Fatores

Para facilitar a automatização do processo de escolha da atividade mais adequada, propomos um modelo que associa a cada conjunto de atividades alternativas, um conjunto de fatores a serem considerados no momento da escolha da atividade que vai ser executada. O tempo de execução, o custo computacional e a qualidade dos dados gerados são exemplos de fatores que podem estar associados para determinar um conjunto de atividades alternativas.

O modelo de fatores é inspirado nos trabalhos de Boeres *et al.* (2011) e Oliveira *et al.* (2012) e é projetado de forma a permitir a ponderação dos fatores de um conjunto de atividades. Cada variabilidade de uma atividade possui v_1, \dots, v_n valores referentes aos n fatores com pesos w_1, \dots, w_n tal que $\sum w_i = 1$. A média ponderada dos fatores de uma atividade é $w_1.v_1 + w_2.v_2 + \dots + w_n.v_n$. 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 são iguais. Por exemplo, se os fatores forem desempenho e qualidade, uma variabilidade pode ter melhor desempenho com qualidade pior enquanto que uma segunda pode ter melhor qualidade com desempenho pior. 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 atividade mais adequada a ser executada naquele momento. A tarefa de atribuir pesos aos fatores é um problema que pode requerer mecanismos de apoio à decisão. A fim de apoiar o cientista nesta tarefa, a abordagem DynAdapt sugere a utilização de métodos como o Promethee o MAUT e o AHP conforme detalhado por Boeres *et al.* (2011). Porém para os experimentos considerados neste artigo, tais métodos não foram utilizados, uma vez que se trata de um experimento com um *workflow* sintético, visando resultados relativos ao tempo total de execução do *workflow* com a troca de atividade.

3.2. O Agente de Adaptação

O Agente de Adaptação é o componente responsável por consultar os valores dos fatores, seus pesos e eleger qual variabilidade deve ser executada. Uma vez escolhida a atividade, o agente de adaptação altera as instâncias que estão em execução. Portanto é preciso que o agente de adaptação tenha permissão para alterar as informações utilizadas pela máquina de *workflow* responsável pela execução do *workflow*. Este componente é chamado quando um dos seguintes eventos ocorrer: (i) inclusão ou atualização do *workflow* na base de proveniência; (ii) inclusão ou atualização em alguma informação no modelo de fatores. Para cada variabilidade, o agente de adaptação executa o seguinte cálculo: $C = \sum_{v_i} w_i \times v_i$. 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 a 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.* consomem os mesmos dados e parâmetros e produzem os mesmos dados e parâmetros. No contexto do SGWfC Chiron,

utilizado no estudo de caso, 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{S} , a atividade alternativa deve seguir os esquemas \mathcal{R} e \mathcal{S} também. Embora a restrição apresentada pareça alta, 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{S} , a relação é análoga. Se as atividades seguintes do *workflow* não estiverem preparadas para consumir o esquema \mathcal{S} (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{S} . Estas questões dizem respeito à concepção do workflow ou da linha de experimento.

3.3. 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*. Estas informações precisam estar parametrizadas no banco de dados e não estáticas no plano de execução do workflow, de forma que tanto a máquina de execução quanto o agente de adaptação possam interagir com elas. Para o estudo de caso deste trabalho 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.

3.4. O Agente de Composição

O agente de composição é o componente responsável por fornecer ao cientista uma interface para a definição do *workflow* e para a manutenção das informações do modelo de fatores. A ideia é que o cientista possa, com esta interface, alterar os pesos e valores dos fatores e a partir destas alterações o agente de adaptação se encarrega de decidir, de acordo com o modelo de fatores, as atividades mais adequadas para serem executadas. Para o escopo deste trabalho utilizamos um conjunto de consultas *template* para manter os valores e pesos do modelo ponderado de fatores. Porém, este agente pode ser estendido para uma interface gráfica com a qual o cientista possa controlar os valores de cada fator e seus respectivos pesos e realizar consultas à base de proveniência para obter informações sobre o estado da execução do workflow. Além disso, alertas sobre o estado da execução podem ser enviados de modo assíncrono para o cientista, de forma que ele não precise manter uma seção aberta durante a execução do workflow, que pode levar dias ou até meses.

4. Avaliação Experimental

Para avaliar o funcionamento da arquitetura proposta, desenvolvemos o DynAdapt em Java e utilizamos a máquina de *workflow* Chiron com o seu respectivo repositório de proveniência armazenado no PostgreSQL. O esquema da base de proveniência foi estendido para acomodar o modelo de Fatores apresentado na Seção 4.1.2. Realizamos uma varredura de parâmetros com um *workflow* sintético W composto por três atividades: A , B e C . Cada instância de cada atividade (tarefa) dura em média 60 segundos com desvio padrão de 5 segundos. A atividade B possui duas variabilidades, B_1 e B_2 . Ao conjunto de atividades $\{B, B_1, B_2\}$ foram associados 3 fatores fictícios: custo, confiabilidade, qualidade. Cada fator é qualificado em uma escala de zero a dez: a atividade B tem custo 8, confiabilidade 6 e qualidade 7; a atividade B_1 tem custo 4, confiabilidade 9 e qualidade 8; e a atividade B_2 tem

custo 6, confiabilidade 5 e qualidade 10. Executar o *workflow* com B oferece o melhor custo, com B₁ produz a melhor confiabilidade e com B₂ gera resultados com melhor qualidade. O fator mais valorizado é a qualidade, com peso 0,4, enquanto custo e qualidade têm peso 0,3. Portanto, baseado nos fatores e pesos, a atividade mais adequada para execução é B₂.

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*, que neste caso é a troca de B por B₂, 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 B₂.

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* e no outro cenário, os resultados parciais estão disponíveis a 60%. 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 3 apresenta os resultados obtidos com ambos os cenários.

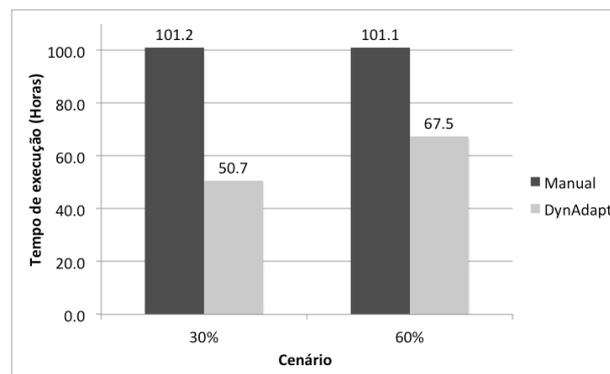


Figura 3. Tempo total de execução comparando as duas abordagens

Para o cenário de 30%, a atividade B foi trocada por B₂ durante a execução da atividade A, ou seja, antes de B ser iniciada. Observamos que neste caso o uso do DynAdapt gerou uma economia de 49,8% de tempo em relação à abordagem manual. Já para o cenário de 60%, a atividade B foi trocada por B₂ durante a execução de B. Neste caso, as tarefas de B que já estavam em execução foram finalizadas e as tarefas que ainda não tinham sido iniciadas foram abortadas. Em seguida todas as tarefas de B₂ são executadas. Observamos que neste caso o DynAdapt reduziu o tempo total de execução em 33,2%. Em ambos os cenários, o uso do DynAdapt reduziu o tempo total da execução do experimento. A redução do tempo foi maior no cenário 30% pois, no momento da troca, nenhuma tarefa de B tinha sido iniciada e, portanto, não há a sobrecarga da finalização de tarefas pendentes nem a re-execução de tarefas. Para o cenário 60%, no momento da troca há tarefas de B que já estavam em execução. Sendo assim, é necessário esperar a finalização destas tarefas para dar prosseguimento à execução do *workflow* com a atividade B₂.

5. Conclusões

Até que se alcance o resultado desejado de um experimento, é comum que os *workflows* científicos passem por adaptações na sua estrutura, como a troca de uma atividade, que pode ter implementações alternativas. Dependendo do ambiente de execução e dos aspectos mais valorizados pelo cientista no momento da execução, uma alternativa pode ser mais vantajosa do que outra. Diante deste cenário, apresentamos o DynAdapt, uma abordagem para permitir que as trocas de atividades por uma de suas alternativas sejam feitas dinamicamente, sem a necessidade de re-executar todo o *workflow* novamente. DynAdapt utiliza um conjunto de pesos e fatores para selecionar a melhor atividade a ser executada no momento e pode trocar uma atividade do workflow por outra em tempo de execução. Os resultados apresentam uma economia de até 49,8% no tempo do experimento desde o início da execução do *workflow* até a obtenção do resultado final.

Agradecimentos. Os autores gostariam de agradecer ao CNPq, Capes e FAPERJ pelo financiamento parcial deste trabalho.

Referências

- 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.
- Clarence Ellis, Karim Keddara, Grzegorz Rozenberg, (1995), "Dynamic change within workflow systems". In: *COCS '95 Proceedings of conference on Organizational computing systems*, p. 10 – 21, New York, NY, USA.
- Costa, F., Silva, V., Oliveira, D., Ocana, K., Dias, J., Ogasawara, E., Mattoso, M., (2013), "Capturing and Querying Workflow Runtime Provenance with PROV: a Practical Approach". In: *International Workshop on Managing and Querying Provenance Data at Scale (BigProv'13)*, Genova, Italy.
- Deelman, E., Gannon, D., Shields, M., Taylor, I., (2009), "Workflows and e-Science: An overview of workflow system features and capabilities", *Future Generation Computer Systems*, v. 25, n. 5, p. 528–540.
- Dias, J., Ogasawara, E., Oliveira, D., Porto, F., Coutinho, A., Mattoso, M., (2011), "Supporting Dynamic Parameter Sweep in Adaptive and User-Steered Workflow". In: *6th Workshop on Workflows in Support of Large-Scale Science*, p. 31–36, Seattle, WA, USA.
- Freire, J., Koop, D., Santos, E., Silva, C. T., (2008), "Provenance for Computational Tasks: A Survey", *Computing in Science and Engineering*, v.10, n. 3, p. 11–21.
- 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.
- Kammer, P. J., Bolcer, G. A., Taylor, R. N., Hitomi, A. S., Bergman, M., "Techniques for Supporting Dynamic and Adaptive Workflow", *Computer Supported Cooperative Work (CSCW)*, p. 269–292.
- 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., Dias, J., Ogasawara, E., Mattoso, M., (2011), "Optimizing Phylogenetic Analysis Using SciHm Cloud-based Scientific Workflow". In: *IEEE e-Science 2011*, p. 190–197, Stockholm, Sweden.
- 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*
- Ogasawara, E., Paulino, C., Murta, L., Werner, C., Mattoso, M., (2009), "Experiment Line: Software Reuse in Scientific Workflows". In: *SSDBM 2009*, p. 264–272, New Orleans, Louisiana, USA.
- Oliveira, D., Ocaña, K., Baião, F., Mattoso, M., (2012), "A Provenance-based Adaptive Scheduling Heuristic for Parallel Scientific Workflows in Clouds", *Journal of Grid Computing*, v. 10, n. 3, p. 521–552.
- Tudruj, M., Kopanski, D., Borkowski, J., (2007), "Dynamic Workflow Control with Global States Monitoring". In: *Parallel and Distributed Computing, 2007. ISPDC '07. Sixth International Symposium on*, p. 44, Hagenberg.