Execução de *Workflows* Científicos de Bioinformática na Nuvem: Experiências e Desafios[1]

Silvia Benza, Kary A.C.S Ocaña, Marta Mattoso

PESC/COPPE - Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro - Brasil

{silviabenza, kary, marta}@cos.ufrj.br

Resumo. *O uso de workflows em nuvens de computadores para experimentos de bioinformática em larga escala permite execuções em paralelo e dar apoio à gerência da grande quantidade de dados biológicos. Entretanto, configurar o experimento na nuvem requer um conhecimento do comportamento dos componentes do experimento. Neste artigo apresentam-se as características do perfil de execução de vários workflows de bioinformática para servir de guia sobre o uso, acoplamento e permitir a avalição de benefícios do desenvolvimento desta metodologia de execução na nuvem.*

1. Introdução

A bioinformática está em constante evolução devido às diversas tecnologias da biologia (*i.e.* sequenciamento de nova geração), que geram um grande volume de dados. Uma alternativa para gerenciar programas que geram o fluxo de dados em larga escala é modelar os experimentos como *workflows* científicos e gerenciá-los por meio de Sistemas de Gerência de *Workflows* Científicos (SGWfC) (Freire *et al.* 2008). O uso de SGWfC provê ao cientista um controle sistemático da gerência da execução das atividades, do fluxo de dados e do experimento como um todo. As nuvens de computadores oferecem um ambiente de processamento de alto desempenho promissor devido à sua capacidade elástica, alta disponibilidade e facilidade de uso, motivo pelo qual já estão sendo adotadas por pesquisadores nas diferentes áreas científicas. O objetivo deste artigo é analisar o comportamento de uma série de *workflows* científicos de bioinformática executados em nuvens nos últimos quatro anos, visando a prover um panorama dessas execuções. Apresenta-se uma descrição da composição, execução paralela e análise desses *workflows* para permitir ao bioinformata avaliar o comportamento frente aos seus próprios experimentos e definir sua configuração. O uso intensivo desses *workflows* vem gerando desafios tanto biológicos como computacionais, e podem alavancar futuras abordagens de bioinformática.

2. Materiais e Métodos

Os dados de proveniência (Freire *et al.* 2008) de seis *workflows* de experimentos de bioinformática foram incluídos na análise. Proveniência representa a ancestralidade de um objeto (*i.e.* dados) e fornece documentação importante para preservar esses dados; determinar a sua qualidade e autoria; e reproduzir, interpretar e validar os resultados gerados. Os *workflows* foram executados com o SciCumulus (Oliveira *et al.* 2010) na nuvem AWS (http://aws.amazon.com/). Eles são: (i) SciHmm (Ocaña *et al.* 2011a) -

[1] O trabalho foi financiado pelas agências de fomento brasileiras FAPERJ e CNPq.

genômica comparativa que apresenta atividades leves, (ii) SciPhy (Ocaña *et al*. 2011b) - filogenia, (iii) SciPhylomics (Oliveira *et al*. 2013) - filogenômica e (iv) SciEvol (Ocaña *et al*. 2012) - evolução molecular, com atividades leves e pesadas que podem ser paralelizadas; (v) SciDock - ancoragem e (vi) SciSamma - modelagem molecular, (a ser publicado em HiComb'14) com atividades leves e pesadas mas não todas paralelizáveis. O tipo de máquina usado nas execuções dos *workflows* foi o *large* com exceção do SciHmm (*micro*) e o número de dados de entrada variou entre 200 e 1000 por execução.

O repositório de proveniência do SciCumulus foi consultado via consultas pré-programadas em SQL (Tabela 1). O objetivo foi obter informações a partir dos metadados dos experimentos armazenados no banco. Por exemplo, com a Consulta 2 da Tabela 1 foram recuperados os tempos mínimo, máximo, médio e o desvio padrão das atividades de cada *workflow* executado com um processador de dois núcleos, ignorando aquelas atividades que possuem erro de execução.

Tabela 1. Consultas SQL ao banco de proveniência do SciCumulus

C1: "Identificar erros na execução das atividades dos *workflows*".	**SELECT** t.* **FROM** hworkflow w, hactivity a, hactivation t **WHERE** w.wkfid = a.wkfid **AND** a.actid = t.actid **AND** ids.wkfid = w.wkfid **AND** t.exitstatus = 1 **AND** w.wkfid = **%WORKFLOW ID %** **ORDER BY** t.starttime	
C2: "Recuperar o tempo mínimo, máximo, médio e desvio padrão das atividades dos *workflows* executados em 2 núcleos, ignorando as atividades que possuem erro de execução"	**SELECT** w.tag, a.tag, **min**(**extract** ('**epoch**' **from** (t.endtime-t.starttime))) **AS** mintime, **max**(**extract** ('**epoch**' **from** (t.endtime-t.starttime))) **AS** maxtime, **avg**(**extract** ('**epoch**' **from** (t.endtime-t.starttime))) **AS** avgtime **stddev**(**extract** ('**epoch**' **from** (t.endtime-t.starttime))) **AS** stddevtime **FROM** hworkflow w, hactivity a, hactivation t (**SELECT** w.wkfid **FROM** hworkflow w, hactivity a, hactivation t **WHERE** w.wkfid = a.wkfid **AND** a.actid = t.actid **GROUP BY** w.wkfid **HAVING** **max**(t.processor) =2 **ORDER BY** wkfid) **AS** wkfids	**WHERE** w.wkfid = a.wkfid **AND** a.actid = t.actid **AND** ids.wkfid = w.wkfid **AND** t.exitstatus = 0 **AND** a.status = 'FINISHED' **AND** w.tag = **%WORKFLOW TAG%** **GROUP BY** a.tag, w.tag **ORDER BY** w.tag

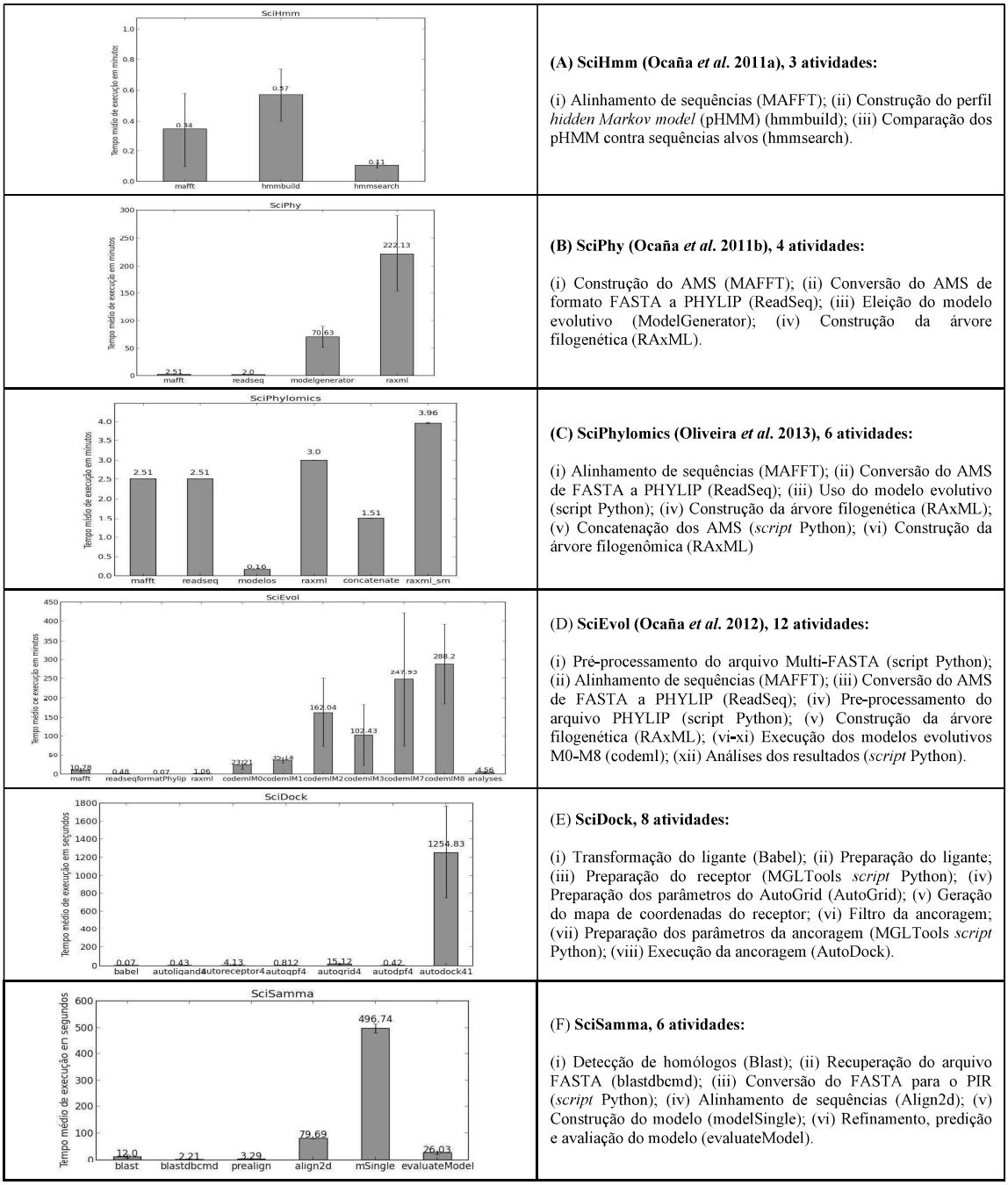
3. Resultados

SciHmm é o *workflow* que possui a menor quantidade de atividades, com o menor tempo médio de execução dentre os seis *workflows* analisados. Estas atividades apresentam um comportamento com execução homogênea, isto é, a diferença do tempo médio entre elas é mínima (Figura 1 (A)), sendo adequado para máquinas *micro* da Amazon. Já SciPhy, SciPhylomics, SciEvol, SciDock e SciSamma (Figura 1 (B-F)), além do aumento no número de atividades, eles apresentam tempos de execução heterogêneos, isto é, a diferença entre os tempos médios é considerável e precisam de máquinas *large*. Os *workflows* apresentam as seguintes características de execução:

- SciHmm, a atividade *hmmbuild* é a que consome mais tempo neste *workflow*.
- SciPhy, as atividades *modelgenerator*, *raxml* são as que requerem mais computação intensiva.
- SciPhylomics, as atividades *raxml* e *raxml_SM* requerem mais computação intensiva, já as atividades *models* e *concatenate* não requerem muito poder de computação intensiva.
- SciEvol, as atividades 1-6 apresentam execução homogênea e requerem baixo poder computacional, enquanto as atividades 7-11 são intensivas, sendo *codemlM8* a mais intensiva.
- SciDock, as primeiras seis atividades apresentam uma execução homogênea e requerem baixo poder computacional, já a última atividade *autodock* requer maior poder computacional.
- SciSamma, todas as atividades apresentam uma distribuição homogênea e requerem baixo poder computacional, só a quinta atividade *modelSingle* requer um maior poder computacional.

Cada um desses *workflows* foi exaustivamente executado na nuvem. Os resultados armazenados e consultados via o banco de proveniência do SciCumulus

foram posteriormente usados para inferir sobre a caraterística dinâmica destes *workflows*, tendo em vista o comportamento heterogêneo das suas atividades. Foi possível determinar uma projeção no valor dos tempos de execução dos *workflows*, pois todas as execuções prévias estão na base de proveniência do SciCumulus. Essa base funciona como um catálogo de estatísticas em Bancos de Dados, desta maneira consultas podem ajudar a estimar o tempo de execução de um novo *workflow* bem como de suas atividades. A média e o desvio padrão do tempo de execução das atividades de todos os *workflows* foram calculados, como apresentado na Figura 1.



(A) SciHmm (Ocaña *et al.* 2011a), 3 atividades:

(i) Alinhamento de sequências (MAFFT); (ii) Construção do perfil *hidden Markov model* (pHMM) (hmmbuild); (iii) Comparação dos pHMM contra sequências alvos (hmmsearch).

(B) SciPhy (Ocaña *et al.* 2011b), 4 atividades:

(i) Construção do AMS (MAFFT); (ii) Conversão do AMS de formato FASTA a PHYLIP (ReadSeq); (iii) Eleição do modelo evolutivo (ModelGenerator); (iv) Construção da árvore filogenética (RAxML).

(C) SciPhylomics (Oliveira *et al.* 2013), 6 atividades:

(i) Alinhamento de sequências (MAFFT); (ii) Conversão do AMS de FASTA a PHYLIP (ReadSeq); (iii) Uso do modelo evolutivo (script Python); (iv) Construção da árvore filogenética (RAxML); (v) Concatenação dos AMS (*script* Python); (vi) Construção da árvore filogenômica (RAxML)

(D) **SciEvol (Ocaña *et al.* 2012), 12 atividades:**

(i) Pré-processamento do arquivo Multi-FASTA (script Python); (ii) Alinhamento de sequências (MAFFT); (iii) Conversão do AMS de FASTA a PHYLIP (ReadSeq); (iv) Pre-processamento do arquivo PHYLIP (script Python); (v) Construção da árvore filogenética (RAxML); (vi-xi) Execução dos modelos evolutivos M0-M8 (codeml); (xii) Análises dos resultados (*script* Python).

(E) **SciDock, 8 atividades:**

(i) Transformação do ligante (Babel); (ii) Preparação do ligante; (iii) Preparação do receptor (MGLTools *script* Python); (iv) Preparação dos parâmetros do AutoGrid (AutoGrid); (v) Geração do mapa de coordenadas do receptor; (vi) Filtro da ancoragem; (vii) Preparação dos parâmetros da ancoragem (MGLTools *script* Python); (viii) Execução da ancoragem (AutoDock).

(F) **SciSamma, 6 atividades:**

(i) Detecção de homólogos (Blast); (ii) Recuperação do arquivo FASTA (blastdbcmd); (iii) Conversão do FASTA para o PIR (*script* Python); (iv) Alinhamento de sequências (Align2d); (v) Construção do modelo (modelSingle); (vi) Refinamento, predição e avaliação do modelo (evaluateModel).

Figura 1. Tempo de execução por atividades dos *workflows* (A) SciHmm, (B) SciPhy, (C), SciPhylomics, (D) SciEvol, (E)SciSamma, (F) SciDock

Cabe mencionar que muitos dos erros foram identificados consultando o banco de proveniência interativamente em tempo de execução, o que permitiu um melhor entendimento desses erros com a intenção de evitá-los em posteriores análises (Consulta 1). Além disso, o SciCumulus permite mudar parâmetros e dados de entrada, ao longo da execução, aproveitando a carga do *workflow* na nuvem.

4. Conclusões e Perspectivas

Apresentamos uma análise exploratória do comportamento de *workflows* científicos de bioinformática executados em nuvens de computador. Os padrões de execução de cada *workflow* mostram variação no que se refere ao poder computacional requerido para cada atividade. Esses resultados evidenciam a importância de o bioinformata ter acesso via consultas aos dados computacionais de execução.

Considerando o SciDock, as primeiras seis atividades requerem pouco poder computacional e poderiam ser alocadas a uma máquina de menor capacidade, enquanto que ao observar que o gargalo está na última atividade, esta deveria ser alocada em uma máquina com capacidade maior. A execução de experimentos desta natureza sugere a alocação dinâmica de máquinas. O cientista pode ter uma ideia prévia do poder computacional necessário e optar por uma única máquina tipo *micro* para as primeiras atividades e por máquinas *large* para realizar a última atividade em paralelo.

A característica dinâmica destes *workflows*, vista por meio do comportamento heterogêneo das atividades, pode beneficiar a composição de novos experimentos similares, para o qual deve ser considerado não somente o fluxo de atividades a serem seguidas, mas também a relação de dependência entre elas. Por tanto, caso o bioinformata opte por executar alguns destes programas ou até *workflows* inteiros, ele poderia prever o comportamento (via comparação) com os apresentados nesta análise. Desta maneira este artigo pode servir como um manual dirigido para aqueles cientistas que ainda não experimentaram o uso da nuvem nos seus experimentos, abrindo com isso, uma nova gama de desafios e oportunidades.

Referências

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.

Ocaña, K., Oliveira, D., Horta, F., Dias, J., Ogasawara, E., Mattoso, M., (2012), Exploring Molecular Evolution Reconstruction Using a Parallel Cloud-based Scientific Workflow. In: *Adv. in Bioinformatics and Computational Biology*, p. 179–191, Berlin, Heidelberg.

Ocaña, K., Oliveira, D., Dias, J., Ogasawara, E., Mattoso, M., (2011a), Optimizing Phylogenetic Analysis Using SciHmm Cloud-based Scientific Workflow. In: *2011 IEEE 7th Inter. Conference on e-Science*, p. 190–197, Stockholm, Sweden.

Ocaña, K., Oliveira, D., Ogasawara, E., Dávila, A., Lima, A., Mattoso, M., (2011b), SciPhy: A Cloud-Based Workflow for Phylogenetic Analysis of Drug Targets in Protozoan Genomes. In: *Adv. in Bioinformatics and Computational Biology*, p. 66–70, Berlin, Heidelberg.

Oliveira, D., Ocaña, K., Ogasawara, E., Dias, J., Gonçalves, J., Baião, F., Mattoso, M., (2013), Performance evaluation of parallel strategies in public clouds: A study with phylogenomic workflows, *Future Generation Computer Systems*, v. 29, n. 7, p. 1816–1825.

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 Inter. Conference on Cloud Computing*, p. 378–385, Washington, DC, USA.