

Serviço de Predição de Recursos para Execução Eficiente de Workflows de Bioinformática em Nuvem Federada com Aprendizado de Máquina

Matheus de C. Sobrinho¹, Aletéia P. F. de Araújo Von Paumgarten¹

¹Departamento de Ciência da Computação - Universidade de Brasília (UnB)

Abstract. *The large amount of resources available across multiple providers in a federation makes it difficult to choose one that is suitable for certain workflows. This work proposes a Machine Learning Resource Prediction Service called sPCRAM. sPCRAM uses a machine learning model combined with a GRASP meta-heuristic to transparently and appropriately size resources, determining the monetary cost and execution time before executing the workflow. The results demonstrate that sPCRAM can adequately estimate the execution time and cost of cloud federation resources on average 97.70% faster than the brute force technique for resource selection.*

Resumo. *A grande quantidade de recursos disponíveis entre vários provedores em uma federação torna difícil escolher um que seja adequado para determinados workflows. Este trabalho propõe um Serviço de Predição de Recursos de Aprendizado de Máquina chamado sPCRAM. O sPCRAM usa um modelo de aprendizado de máquina combinado com uma meta-heurística GRASP para dimensionar os recursos de forma transparente e adequada, determinando o custo monetário e o tempo de execução antes da execução do workflow. Os resultados demonstram que o sPCRAM pode estimar adequadamente o tempo de execução e o custo dos recursos da federação em nuvem em média 97,70% mais rápido do que a técnica de força bruta para seleção de recursos.*

1. Introdução

Computação em nuvem pode ser definido como um paradigma computacional que fornece o modelo de entrega de serviços sob demanda para o usuário final com o intuito de atender às suas necessidades atuais, sem limitações geográficas, com baixo custo, de forma flexível e transparente [Buyya et al. 2010]. As federações em nuvem são conjuntos de provedores que têm todos os seus recursos gerenciados por uma interface padrão, o que possibilita gerenciar as diversas demandas de forma compartilhada com todos os provedores da federação, atendendo a demanda computacional [Rochwerger et al. 2009].

Workflows de bioinformática são complexos e envolvem experimentos que exigem computação significativa devido à grande quantidade de dados usados [Rosa et al. 2021]. Sendo assim, há uma necessidade de serviços de previsão para ajudar os usuários a escolher os recursos certos em uma federação de nuvem. Um serviço de previsão eficiente pode melhorar os custos e reduzir as falhas durante a execução e processamento de um determinado fluxo de trabalho de Bioinformática, evitando que o usuário exija uma infraestrutura superestimada ou subestimada.

O BioNimbuZ é uma plataforma de federação que garante a integração entre diferentes nuvens de forma simples, dinâmica e transparente e entrega um modelo de predição que utiliza métodos estatísticos [Rosa et al. 2021], no que tange a execução de *workflows* científicos de Bioinformática. Esse modelo auxilia o usuário na previsão do tempo de execução, custo e dimensionamento dos recursos para a execução de um fluxo de trabalho usando métricas, meta-heurística e um modelo estatístico. Além dos modelos estatísticos, é possível aplicar o Aprendizado de Máquina para fornecer um processo de previsão de tempo de execução e dimensionamento de recursos mais eficiente, utilizando-o durante a execução de um fluxo de trabalho de Bioinformática.

Assim, este trabalho propõe um Serviço de Predição de Recursos de Aprendizado de Máquina denominado sPCRAM. O sPCRAM pode prever adequadamente a demanda dos recursos computacionais necessários para executar *workflows* de Bioinformática em uma plataforma de nuvem federada com eficiência. Os custos são estimados combinando a estimativa de tempo de execução e a meta-heurística GRASP. O serviço proposto foi integrado na plataforma BioNimbuZ, em sua segunda versão, para auxiliar na predição de recursos da mesma. Os experimentos são baseados em um fluxo de trabalho de RNA-Seq (montagem, mapeamento e anotação) do fungo *Aspergillus fumigatus* e executados em cenários de nuvem federada dos provedores Google Cloud Platform (GCP) e Amazon Web Service (AWS). Este trabalho atualiza a plataforma BioNimbuZ com o serviço de previsão proposto, o sPCRAM. O sPCRAM fornece uma maneira de entender melhor o tempo de execução e as características de desempenho de projetos de bioinformática, usando modelos de aprendizado de máquina supervisionados para problemas de regressão.

No entanto, estimar a quantidade necessária de recursos está longe de ser uma tarefa trivial, pois as execuções se tornam uma caixa preta, com o risco de resultados imprecisos e onerosos [Chaisiri et al. 2011]. Assim, a escolha errada na quantidade ou mesmo na capacidade do recurso pode impactar negativamente a execução do *workflow* e afetar seu custo monetário [Rosa et al. 2021]. No contexto de previsão de recursos, uma abordagem que se destaca na literatura é a meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) [Feo and Resende 1995]. O GRASP é um processo iterativo de duas fases (construção e pesquisa local) no qual as iterações são independentes. A fase de construção aponta uma solução inicial que a fase de busca local pode melhorar.

Assim, este trabalho propõe uma nova abordagem de conselho de preditores, juntamente com a meta-heurística GRASP, para estimar o tempo e os recursos necessários para executar *workflows* de Bioinformática no ambiente de orquestração de nuvens do BioNimbuZ versão 2.

1.1. sPCRAM

O sPCRAM tem como objetivo estimar o tempo de execução e custo para execução de *workflows* de Bioinformática. O sPCRAM considera os interesses dos usuários para permitir a alocação de recursos de nuvem federados alinhados com finalidades como execução de baixo custo e um tempo de execução mais longo ou uma execução de alto custo e um tempo de execução mais curto.

O Conselho de Preditores compreende um modelo de regressão multilinear, uma rede neural *perceptron* multicamadas, no inglês, *Multilayer Perceptron* (MLP) e uma flo-

resta aleatória. As saídas desses modelos servem de base para o treinamento da rede neural MLP, que define o Conselho de Preditores. As previsões anteriores dos modelos são necessárias para filtrar e obter a previsão final. Desta forma, é possível filtrar e minimizar os erros de previsão de cada modelo individualmente pelos modelos previstos antes da entrada do Conselho de Preditores. O modelo de conselho de previsão é definido como uma Rede Neural MLP, desenvolvida com o componente *Scikit-learn* da biblioteca [Pedregosa et al. 2011] *MLRegressor*. A meta-heurística GRASP compreende a fase de construção e a busca local. Uma solução encontrada na fase de construção pode ser melhorada na fase de busca local. Uma notação e uma definição de uma função de custo, também conhecida como função objetivo, pode ser vista na Equação 1. A meta-heurística GRASP recebe a lista de recursos disponíveis, os detalhes do *workflow*, contendo os elementos necessários à execução, bem como as restrições dos usuários para a previsão dos recursos.

Assim, uma solução $s \in S$ é viável se s não violar os requisitos do usuário definidos durante a fase de coleta de informações. O tempo máximo $tp(s)$ é o tempo total de execução estimado pelo modelo preditivo para s e c_{udd} como o custo de *upload*, *download* e armazenamento de dados em uma máquina virtual p . O valor da máquina virtual é descrito como c_p , os desejos do usuário relacionados ao custo e tempo são definidos como C_m e T_m , respectivamente.

$$F(s) = \left(\alpha_1(c_p + c_{udd}) + \alpha_2tp(s) \right) + \lambda_1 \left(abs(tp(s) - T_m) \right) + \lambda_2 \left(abs(c_p - C_m) \right) \quad (1)$$

A função de custo descrita na Equação 1 calcula a qualidade da solução atual e aponta a necessidade de troca da melhor solução encontrada até o momento. Esta função visa minimizar os custos financeiros e o tempo total de execução de um *workflow*, relaxar as desigualdades e penalizar a diferença entre o tempo máximo e o custo monetário inicialmente especificado pelo usuário. Os coeficientes de penalidade associados à violação dos requisitos de tempo, λ_1 , e o custo, λ_2 , são empregados como a diferença absoluta entre o coeficiente e o custo ou o tempo. Vale ressaltar que esta função de custo não considera o tempo de alocação das máquinas virtuais. Dessa forma, a solução lida com o tempo necessário para a execução do software e estima os custos do fluxo de trabalho por meio de seu conjunto de programas antes de sua execução real.

2. Resultados

Esta seção apresenta os resultados de sPCRAM ao executar um fluxo de trabalho de Bioinformática em um ambiente de nuvem federado por meio de um *plugin* desenvolvido para a plataforma BioNimBuZ. Os resultados apresentados demonstram o comportamento do sPCRAM nas previsões de custo monetário e tempo de execução.

Usamos os provedores de nuvem *Google Cloud Platform (GCP)* e *Amazon Web Services (AWS)*, ambos em sua opção de recursos gratuitos por tempo limitado como microserviços BioNimbuZ. As máquinas preventivas não foram consideradas. As máquinas variaram no número de núcleos de processamento, bem como na capacidade de RAM.

A solução proposta neste trabalho foi avaliada por meio de experimentos realizados em diferentes cenários, em execuções consideradas de alto custo e de baixo custo. As

colunas α_1 e α_2 representam os pesos da função objetivo de custo e tempo, respectivamente.

2.1. Workflow Executado

O RNA-Seq é um experimento para traçar o perfil da expressão diferencial do gene, identificando mudanças nos transcriptomas celulares ou teciduais. Foi estruturado um fluxo de trabalho de RNA-Seq usando *Aspergillus fumigatus* cepa 293 leituras de extremidade única sequenciadas com Illumina HiSeq 2000. O experimento original foi projetado para identificar genes diferencialmente expressos (DEGs) de *A. fumigatus* 293 sob os efeitos da metilprednisolona. Neste fluxo de trabalho, os softwares de Bioinformática empregados foram o Prinseq para filtragem; Bowtie e TopHat para mapeamento e junção de emenda, respectivamente; Trinity para montagem; e Blast para anotação.

Foram executadas várias instâncias do mesmo *workflow* em diferentes provedores de nuvem, capturando os dados de execução pretendidos para construir um banco de dados de treinamento que foi então usado como entrada para os três modelos de aprendizado de máquina desenvolvidos que definem as estimativas de custo e tempo.

2.2. Resultados do Conselho de Preditores

No cenário 1, o qual o software é conhecido na base de dados, o conselho de preditores pode estimar o tempo de execução dos programas, que fazem parte do fluxo de trabalho, que estão na base de conhecimento construída. Foi calculada uma aproximação com um erro médio de 3 segundos para TopHat2, 25 segundos para Trinity e 170 segundos para BlastX. Considerando o tempo total de execução do *workflow* de experimento, o erro pode ser considerado aceitável [Rosa et al. 2021].

No cenário 2, o qual o software é desconhecido na base de dados, o sPCRAM não tem uma base histórica de tempo de execução para os programas presentes no fluxo de trabalho. Execuções simuladas propostas em [Rosa et al. 2021] foram utilizadas como medida de contorno nos casos em que o modelo não conhece detalhes da execução do programa que se destina a prever o tempo. Portanto, as execuções simuladas são realizadas com uma pequena parte do arquivo de entrada para reduzir o erro de previsão. No cenário 2 o modelo preditivo obteve um erro médio de 8 segundos, permanecendo próximo ao apresentado em [Rosa et al. 2021].

2.3. Resultados da Meta-heurística GRASP

Os testes foram realizados com os programas que compõem o *workflow* de Bioinformática apresentado. A seleção de recursos sPCRAM busca selecionar a melhor solução entre todas as instâncias disponíveis no ambiente de federação em nuvem. Implementamos uma técnica de força bruta e a meta-heurística GRASP para comparar e estabelecer um nível de qualidade para a seleção de recursos. Outras técnicas podem ser exploradas em trabalhos futuros, como por exemplo, os algoritmos genéticos. A técnica de força bruta utiliza a mesma função objetivo que a meta-heurística GRASP, mas percorre todas as máquinas disponíveis naquele ambiente. Os resultados mostraram que em todos os cenários, a meta-heurística GRASP é, em média, 97,70% mais rápida que a técnica de Força Bruta para os casos avaliados, e com uma diferença percentual em relação ao custo financeiro de 2,2% entre GRASP e Força Bruta para os cenários de execução avaliados.

Nesse sentido, é possível afirmar que o serviço proposto neste trabalho é adequado para uso, no que se refere à previsão de tempo e recursos para *workflows* científicos em um ambiente de nuvem federada, como era possível, a partir da lista de recursos de ambos os provedores usados para teste, para estimar tempo e recursos adequados para os cenários apresentados. A seção seguinte apresenta os trabalhos relacionados e contrasta alguns deles com a proposta deste trabalho.

3. Conclusão

Este trabalho apresentou um Serviço de Predição de Recursos de Aprendizado de Máquina denominado sPCRAM. O sPCRAM permite uma seleção eficiente de recursos para *workflows* de Bioinformática em uma nuvem federada por meio de uma meta-heurística GRASP para o melhor custo-benefício dos usuários. O sPCRAM pode prever um tempo de execução do fluxo de trabalho de Bioinformática usando uma base de programa conhecida e uma estrutura de limite para os programas desconhecidos. Ele foi testado usando os dois tipos mais comuns de *workflows* de RNA-Seq: de novo e baseado em mapeamento.

O Serviço de Predição proposto neste trabalho pode estimar recursos para um fluxo de trabalho conforme apresentado nos estudos de caso ajustando estimativas aos cenários definidos pelo usuário. Ele comprova a premissa inicial de evitar o desperdício de recursos e apoiar as escolhas de recursos para os usuários. Trabalhos futuros pretendem explorar outras técnicas de seleção de recursos, como algoritmos genéticos, ambientes de programação paralela, e expandir os tipos de *workflows* de Bioinformática para Genômica, Célula Única e outras análises. Essas melhorias incluem a implementação de um banco de dados distribuído integrado à plataforma BioNimbuZ para melhorar o acesso às informações do banco de dados histórico.

Referências

- Buyya, R., Ranjan, R., and Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 13–31. Springer.
- Chaisiri, S., Lee, B.-S., and Niyato, D. (2011). Optimization of resource provisioning cost in cloud computing. *IEEE transactions on services Computing*, 5(2):164–177.
- Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.
- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., et al. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4–1.
- Rosa, M. J., Ralha, C. G., Holanda, M., and Araujo, A. P. (2021). Computational resource and cost prediction service for scientific workflows in federated clouds. *Future Generation Computer Systems*, 125(2):844–858.