

Escalonamento de workflows científicos ciente de energia verde em servidores distribuídos

Guilherme Almeida¹, Marcelo T. do Ó¹, Emilio Francesquini², Daniel Cordeiro^{1*}

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brazil

²Centro de Matemática, Computação e Cognição – Universidade Federal do ABC
Santo André – SP – Brazil

{guilherme.augusto.almeida, marcelo.torres.o, daniel.cordeiro}@usp.br
e.francesquini@ufabc.edu.br

Abstract. *Using distributed computing, tasks in a scientific workflow can be scheduled across several servers. The current energy consumption of servers is very high, and much of this energy is generated from non-renewable sources. This study proposes the development of a green energy-aware scheduling algorithm, which will allow tasks to be performed more sustainably.*

Resumo. *Com o uso da computação distribuída, as tarefas de um workflow científico podem ser escalonadas em vários servidores. O consumo atual de energia em servidores é bastante alto e muita desta energia é gerada por fontes não-renováveis. Este estudo propõe o desenvolvimento de um algoritmo de escalonamento ciente de energia verde, que permitirá que as tarefas sejam executadas de forma mais sustentável.*

1. Introdução

Os *workflows* científicos são aplicações que geralmente trabalham com um grande volume de dados e tendem a ser computacionalmente intensivos (*e.g.* simulações de fenômenos complexos). Nesse tipo de aplicação, o processo é descrito como um fluxo de atividades no qual são estabelecidas relações de dependência entre as tarefas [Oda et al. 2018].

Com a disponibilidade da computação distribuída, seja ela feita em servidores aglomerados (*clusters*) ou em nuvens *cloud*, as tarefas de um *workflow* científico podem ser distribuídas para execução em vários servidores. Entretanto, o consumo de energia em servidores de nuvem é bastante alto e tende a continuar crescendo com a maior adoção do processamento distribuído ou que faça uso desta plataforma [Buyya et al. 2018].

Além disso, uma grande parcela da energia utilizada por estes servidores é gerada a partir de fontes não-renováveis, contribuindo para um impacto ambiental negativo ainda maior [Buyya et al. 2018]. Contudo, há diversas iniciativas para fazer com que a energia utilizada por alguns servidores passe a ser produzida por fontes renováveis, como solar e eólica, o que tende a diminuir este impacto ambiental [Gupta et al. 2019].

*Este trabalho foi financiado pelos projetos nº 2019/26702-8 e 2021/06867-2, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Workflows científicos são aplicações que requerem maior poder computacional, bem como maiores custos e tempo de execução. O estudo de [Niaz et al. 2022] sugere que usar apenas energia de fontes renováveis em servidores é mais caro, enquanto usar uma combinação entre energia renovável (verde) e energia não-renovável (marrom) é mais econômico. Portanto, um escalonador de tarefas que leve em conta a disponibilidade de energia verde nos servidores pode desempenhar um papel importante na redução de impactos ambientais e custos financeiros na execução de *workflows* científicos.

Embora a energia verde apresente vantagens, existem desafios relacionados à sua intermitência. Por exemplo, no caso das fontes de energia eólica e solar a produção depende, respectivamente, da intensidade dos ventos e da irradiação solar. Além disso, os servidores podem estar distribuídos geograficamente em cidades ou países distintos, o que aumenta a variação de disponibilidade de energia verde. Por isso são necessárias estratégias para que haja um melhor aproveitamento desse tipo de energia.

A abordagem de escalonamento *follow-the-renewables* consiste em alocar tarefas em servidores com maior disponibilidade de energia verde [Vasconcelos et al. 2022]. Este artigo descreve um projeto de pesquisa em andamento cujo objetivo é investigar técnicas de *follow-the-renewables* e implementar um escalonador multiobjetivo que diminua a emissão de CO₂ durante a execução de um *workflow* científico em um *cluster* no qual os servidores possuem diferentes disponibilidades de energia verde.

2. Materiais e métodos

O escalonador será implementado em um *framework* de paralelismo já existente: Omp-Cluster (OMPC) [Yviquel et al. 2022]. O OMPC permite com que um conjunto de processos MPI seja definido como *target device* para *tasks* OpenMP.

2.1. OpenMP e MPI

O OpenMP é uma API para simplificar o desenvolvimento de aplicações *multithreading* através da adição de diretivas no código fonte. As *tasks*, introduzidas na versão 3.0, são instâncias específicas de código executável e seus ambientes de dados. A partir de sua versão 4.0, o OpenMP passou a permitir o mapeamento de variáveis para ambientes de dados de *devices* e a executar operações nestes, definindo-os como *target devices* [OpenMP Architecture Review Board 2021].

Essa técnica, conhecida como *computation offloading*, permite transferir o processamento para um dispositivo especializado na tarefa, tal como uma *Graphic Processing Unit* (GPU). Entretanto, nas implementações tradicionais do OpenMP esta técnica está limitada a dispositivos presentes no mesmo nó [Yviquel et al. 2022].

Por sua vez, *Message-Passing Interface* (MPI) é uma interface para a paralelização de programas distribuídos e, diferentemente das *threads* do OpenMP, opera em diferentes memórias e, por isso, trabalha com o envio e recebimento de mensagens entre processos [Pacheco 2011]. Além disso, o MPI pode ser utilizado em programas *multi-thread*, porém, o “nível de *thread*” permitido pode variar entre somente uma *thread* a executar por vez até múltiplas *threads* permitidas fazer chamadas MPI em paralelo sem restrições de concorrência (MPI_THREAD_MULTIPLE) [Message Passing Interface Forum 2021].

2.2. OMPC

O OMPC é uma extensão do OpenMP para aplicar o paralelismo de tarefas em *clusters*. Dessa forma, é possível utilizar as diretivas do OpenMP para criar programas paralelos em um *cluster* de alto desempenho, aumentando a quantidade de recursos computacionais disponíveis [Yviquel et al. 2022].

O programador determina o *workflow* estabelecendo a relação de dependência entre as tarefas e os dados através da declaração de diretivas OpenMP. Assim, o escalonador consegue distribuir os dados e as tarefas de forma otimizada. A comunicação entre os nós é feita usando uma implementação de MPI de forma transparente para o desenvolvedor. Esta comunicação é realizada utilizando as definições de *task* e *target device* do OpenMP.

Um dos conjuntos de componentes do OMPC são escalonadores para distribuição de tarefas. Atualmente estão implementados algoritmos *round-robin* e HEFT [Topcuoglu et al. 2002]. Este conjunto de escalonadores pode ser expandido com novas implementações que sigam o padrão estabelecido para o OMPC, uma vez que seu código é *open source* e novos escalonadores podem ser adicionados sem maiores alterações nos demais componentes.

3. Escalonador Ciente de Energia Verde

Este estudo tem como objetivo a implementação de um algoritmo de escalonamento multiobjetivo, com o propósito de diminuir as emissões de CO₂ ao mesmo tempo que mantenha de forma equilibrada o tempo de execução e os custos financeiros associados.

Uma das possibilidades para o algoritmo será uma implementação de variação do HEFT [Topcuoglu et al. 2002]. O HEFT tem como variáveis de entrada o conjunto de *tasks* representado por um grafo acíclico dirigido, os processadores disponíveis, os tempos de processamento de cada *task* em cada processador e os tempos de transmissão de dados entre as *tasks* em diferentes processadores. Seu objetivo é diminuir o tempo final de execução.

Além destes fatores, o novo algoritmo deve também considerar a disponibilidade de energia verde e da rede de abastecimento, bem como a situação do *cluster* no momento. Ainda que seja variável, a energia solar é mais previsível do que a energia eólica. Por isso, esse tipo de energia será considerado no escopo do presente trabalho.

Em [Vasconcelos et al. 2023] demonstrou-se que é possível reduzir de forma eficiente o consumo de CO₂ em um provedor de nuvem distribuído geograficamente ao redor do planeta, fazendo uso de energia produzida por painéis fotovoltaicos (PV) e da rede de abastecimento. O modelo também considerou o uso de baterias para armazenar energia dos PVs, pois a produção varia de acordo com a hora do dia, estação do ano e fatores climáticos. O uso de baterias também pode ser considerado no presente trabalho.

Os tipos de aplicação a serem escalonados por essa abordagem dependerão dos resultados preliminares obtidos através da experimentação com alguns algoritmos. Serão estudados algoritmos presentes na literatura sobre escalonamento de *workflows* científicos cientes de energia.

A implementação e execução do algoritmo de escalonamento proposto e da carga escolhida será feita no ambiente do OMPC. A disponibilidade de energia verde será simulada com base nos dados disponibilizados por [Gelaro et al. 2017].

Com o desenvolvimento e uso deste algoritmo de escalonamento ciente de energia verde, aplicações poderão ser executadas de forma mais sustentável, através de implementações como a apresentada pelo nosso estudo.

References

- Buyya, R. et al. (2018). A manifesto for future generation cloud computing: Research directions for the next decade. *ACM Comput. Surv.*, 51(5).
- Gelaro, R. et al. (2017). The modern-era retrospective analysis for research and applications, version 2 (merra-2). *Journal of Climate*, 30(14):5419 – 5454.
- Gupta, V., Shenoy, P., and Sitaraman, R. K. (2019). Combining renewable solar and open air cooling for greening internet-scale distributed networks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy '19*, page 303–314, New York, NY, USA. Association for Computing Machinery.
- Message Passing Interface Forum (2021). MPI: A message-passing interface standard version 4.0.
- Niaz, H., Shams, M. H., Zarei, M., and Liu, J. J. (2022). Leveraging renewable over-supply using a chance-constrained optimization approach for a sustainable datacenter and hydrogen refueling station: Case study of california. *Journal of Power Sources*, 540:231558.
- Oda, R., Cordeiro, D., and Braghetto, K. R. (2018). Dynamic resource provisioning for scientific workflow executions in clouds. In *2018 IEEE International Conference on Services Computing (SCC)*, pages 291–294.
- OpenMP Architecture Review Board (2021). OpenMP application program interface version 5.2.
- Pacheco, P. (2011). *An Introduction to Parallel Programming*. Morgan Kaufmann.
- Topcuoglu, H., Hariri, S., and Wu, M.-Y. (2002). Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274.
- Vasconcelos, M., Cordeiro, D., Costa, G. D., Dufossé, F., Nicod, J.-M., and Rehn-Sonigo, V. (2023). Optimal sizing of a globally distributed low carbon cloud federation. In *The 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*. No prelo.
- Vasconcelos, M. F. S., Cordeiro, D., and Dufossé, F. (2022). Indirect network impact on the energy consumption in multi-clouds for follow-the-renewables approaches. In *Proceedings of the 11th International Conference on Smart Cities and Green ICT Systems – SMARTGREENS*, pages 44–55. INSTICC, SciTePress.
- Yviquel, H., Pereira, M., Francesquini, E., Valarini, G., Leite, G., Rosso, P., Ceccato, R., Cusihualpa, C., Dias, V., Rigo, S., Souza, A., and Araujo, G. (2022). The OpenMP cluster programming model. In *Workshop Proceedings of the 51st International Conference on Parallel Processing*. ACM.