

MODELAGEM DE TRABALHOS COMUNICANTES PARA EXECUÇÃO DE ALGORITMOS APRENDIZADO DE MÁQUINA¹

Guilherme Diel², Guilherme Piegas Koslovski³.

¹ Vinculado ao projeto “Mecanismos para Alocação de Infraestruturas Virtuais Baseados em Aprendizado de Máquina e Acelerados por GPUs”

² Acadêmico do Curso de Ciência da Computação – CCT – Bolsista PROBIC/UDESC

³ Orientador, Departamento de Ciências Tecnológicas – CCT – guilherme.koslovski@udesc.br

⁴ Acadêmico do Curso de Ciência da Computação – CCT

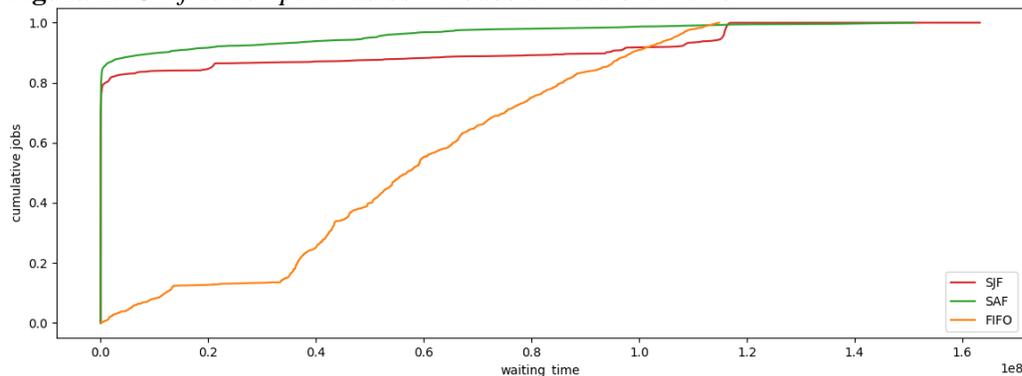
High Performance Computing (Computação de Alto Desempenho), é essencial para o mundo hoje em dia, devido às diversas aplicações em pode-se ser aplicado, indo desde simulações de dinâmica de fluidos até processamento de transações[2]. Dessa forma, achar métodos de tornar esses processos mais rápidos e eficientes se torna muito relevante atualmente. Esse tipo de tecnologia é geralmente usado em *clusters* (rede de computadores que trabalham junto, como sendo um único), sendo aplicado *Parallel Computing* (Computação Paralela), que consiste em resolver processos simultaneamente, sendo ideal para resolver grandes problemas. Além disso, para melhor desempenho do cluster em HPC, podem ser usados diversos métodos de escalonamento, que é um processo em que as tarefas a serem processadas pelo cluster são organizadas de tal maneira que a necessidade do pesquisador seja mais bem atendida. Por exemplo, caso alguém necessite garantia de que todas as tarefas serão processadas (não haja *starvation*) é indicado FIFO (*First In First Out*) em que os processos são processados conforme vão chegando na fila[3]. Entretanto, caso não seja necessário garantir que não haja *starvation*, há muitos métodos de escalonamento, como SPF (*Shorrtest Processing time Fisrt*), o qual coloca o processo com menor tempo de processamento primeiro e SAF (*Shortest Area First*), em que o processo com o menor produto entre o tempo de processamento e o recurso utilizado é colocado no início da fila[1].

Assim, neste projeto, foram estudados três tipos de escalonadores em um ambiente que simula eventos discretos, em que o tempo é organizado e processado sequencialmente e as requisições são enviadas de forma on-line e sem dependências. Com o intuito de aprender sobre os diversos métodos de escalonamento e os benefícios específicos de cada método diferente, para conseguir aplicá-los em cenários da vida real e simular como cada método iria performar, foram testados os escalonamentos FIFO (*First In First Out*), SPF (*Shortest Processing time First*) e SAF (*Shortest Area First*). Inicialmente foram gerados casos de testes aleatórios, entretanto, visando obter resultados mais próximos de um cenário real, foi utilizado Rastros de execução de sistemas de produção de processos processados paralelamente documentados e bem reconhecidos. O nome do rastro de execução utilizado é SDSC SP2, que conta com, aproximadamente, 73.000 processos. Dessa forma, um processo é uma tarefa ou operação que deve ser processada pelo programa ou sistema operacional. Os processos utilizados, possuem 18 atributos, das quais foram levadas em conta, para o escalonamento, apenas quatro: nome, tempo de chegada, tempo de processamento e recurso que necessita.

Apesar de não ser uma tarefa fácil especificar um melhor método de escalonamento, visto que depende do objetivo final de cada pesquisador, o método que apresentou uma média de tempo de espera (*waiting time*) entre ser colocado na fila até realmente ser escalonado foi o SAF, em segundo lugar foi o SPF e por último o FIFO. Apesar de o SPF ter uma média de *waiting time* melhor que o FIFO, esse foi o método que obteve trabalhos com maior *waiting time* dentre os estudados.

A Figura 1 apresenta um gráfico do tipo diagrama de fluxo cumulativo (CFD), de todos os métodos de escalonamento estudados, em que o eixo Y vai de 0 a 1 e o eixo X representa o tempo de espera de cada processo (momento que foi escalonado – momento que chegou na fila). Dessa forma, é possível perceber que SAF teve menor área, seguido por SJF e FIFO. Entretanto, também é possível notar que apenas com FIFO é garantido que não haverá starvation, visto que com mais jobs, SAF e SJF aumentariam exponencialmente, não sendo garantido que todos os jobs sejam devidamente processados.

Figura 1. Gráfico comparando os métodos de escalonamento



Palavras-chave: HPC. Escalonadores. Parallel Computing.

Referências:

- D. Carastan-Santos, R. Y. De Camargo, D. Trystram and S. Zrigui, "One Can Only Gain by Replacing EASY Backfilling: A Simple Scheduling Policies Case Study," 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2019, pp. 1-10, doi: 10.1109/CCGRID.2019.00010.SANTOS, D. C. et al. One Can Only Gain By Replacing EASY Backfilling: A simple scheduling policies case study. Disponível em: <https://hal.archives-ouvertes.fr/hal-02237895/document#:~:text=We%20conclude%20that%20one%20can,and%20the%20transparency%20of%20FCFS>. Acesso em 30/08/2022.
- Feitelson, Dror G., Dan Tsafir, and David Krakov. "Experience with using the parallel workloads archive." *Journal of Parallel and Distributed Computing* 74.10 (2014): 2967-2982.
- Feitelson, Dror G., et al. "Theory and practice in parallel job scheduling." *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, Berlin, Heidelberg, 1997.