

Avaliação de Diretivas Paralelas em uma Aplicação de Simulação de Secagem de Grãos*

Natiele Lucca[†], Claudio Schepke, Aline Vieira de Mello, Anna Victória G. Marciano[‡]

¹Laboratório de Estudos Avançados em Computação (LEA)
Universidade Federal do Pampa (UNIPAMPA) – Alegrete – RS – Brazil

natielelucca@gmail.com

Resumo. Este trabalho apresenta uma avaliação de diferentes estratégias de paralelismo usando diretivas de OpenMP em uma aplicação de dinâmica dos fluidos de simulação de meios porosos. Os resultados obtidos mostraram que todas as versões apresentaram ganho de desempenho.

1. Introdução

Neste trabalho, uma aplicação de meios porosos, que simula o processo de secagem de grãos, foi acelerada com o intuito de reduzir o tempo de execução. A implementação original, baseada na discretização por volumes finitos do espaço bidimensional e tempo, foi modificada a fim de avaliar abordagens de paralelismo eficientes e não tão usuais utilizando OpenMP para ambientes multi-core e também OpenACC para GPU [Lucca 2022].

Foram implementadas 4 versões usando diretivas de OpenMP: *parallel do*, *target e teams* para Multi-core, e OpenACC *parallel do* para GPU [Lucca et al. 2023]. As versões OpenMP *target* e OpenACC exigiram alterações na estrutura no código-fonte para paralelizar o código. O desempenho foi avaliado para três tamanhos de malha (51x63, 100x124 e 200x249) e para 2, 4, 8, 16 e 32 threads/teams paralelos. A avaliação dos experimentos foi conduzida em uma workstation com 2 processadores octa-core Xeon E5-2650, com 2.80 GHz (16 cores virtuais) e 128 GB de Memória, e uma GPU Quadro M5000. Para a compilação usou-se *nvfortran* do HPC_SDK toolkit da NVidia, versão 23.1.

2. Resultados

A Figura 1 apresenta a média de 25 execuções dos resultados do tempo das execuções e desvio padrão (valor do topo da coluna) para a matriz 200x249. O tempo sequencial é destacado pela coluna vermelha, seguido pelos valores de OpenMP *do*, OpenACC, OpenMP *target* e *teams*. Os resultados mostram que a implementação é escalável, uma vez que o *speedup* aumenta à medida que um número maior de threads é utilizado.

As versões OpenMP *parallel do* e *teams* apresentaram resultados semelhantes. Os resultados mostram que *teams* e *target* são opções para a abordagem tradicional de paralelismo de *loop* paralelo usada em meios porosos e outras aplicações de demanda de alto desempenho semelhantes. Nos testes realizados ambas versões obtiveram resultados semelhantes. A escalabilidade é limitada ao tamanho de malha.

*Trabalho parcialmente financiado pelo Edital PqG FAPERGS 07/2021: Projeto 21/2551-0002055-5.

[†]Bolsista CAPES PDPG-FAP (Parcerias Estratégicas nos Estados: N° 42046017016P1) 2021 e 2022.

[‡]Bolsista PIBIC/CNPq 2023.

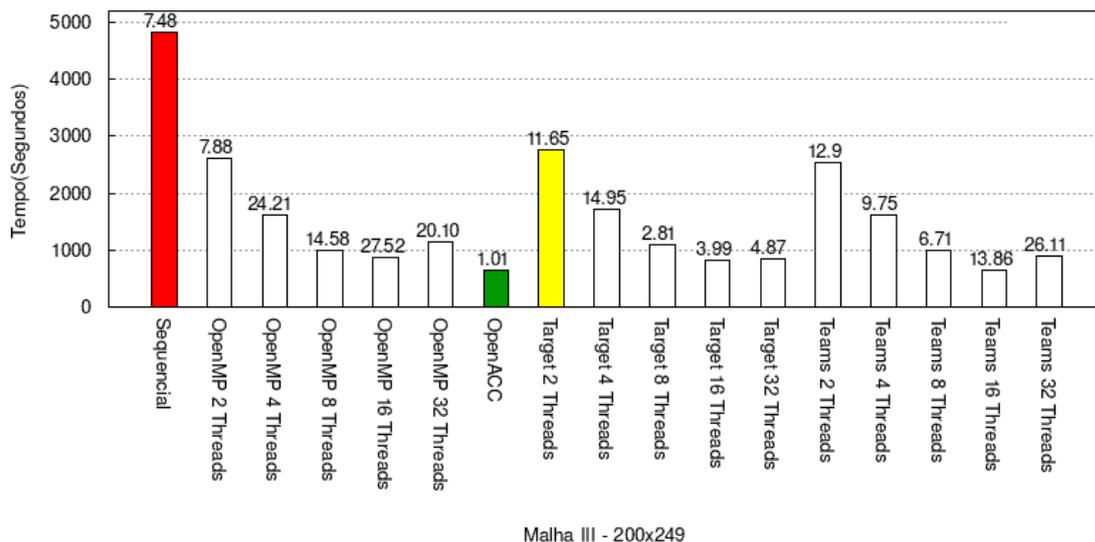


Figura 1. Tempo de Execução - Resultado da malha 200x249

As versões OpenACC e OpenMP *target* obtiveram resultados diferentes. Enquanto o OpenMP *target* foi eficiente para a malha menor, o OpenACC foi eficiente para os testes com a malha maior. Essa discrepância pode ocorrer devido às características do compilador ou a necessidade de aperfeiçoar as diretivas utilizadas para explorar o paralelismo nas versões. A aceleração utilizando uma única GPU foi de 7,4 para o tamanho de malha 200×249 . Consideramos este valor aceitável devido ao tamanho desta malha não explorar todas as capacidades da GPU. O ideal seria utilizar malhas maiores ainda, a fim de maximizar a capacidade de execução que a GPU provê.

Alguns rastros de execução também foram coletados variando o número de threads e o tamanho das malhas [da Silva et al. 2022]. Utilizamos a ferramenta Vampir para visualização de desempenho, ou rastreamento de execução, identificando as etapas paralelas e o tempo de sincronização. Os resultados mostraram os 36 *loops* paralelos chamados em cada etapa de convergência da implementação do OpenMP *parallel* do. A ferramenta permite identificar tempos de espera para sincronização e etapas sequenciais do código. Em trabalhos futuros pretendemos acrescentar a avaliação da interface de programação CUDA. Também pretendemos utilizar a ferramenta *nvprof* para avaliar a performance das APIs que utilizam GPU.

Referências

- da Silva, H. U., Lucca, N., Schepke, C., de Oliveira, D. P., and da Cruz Cristaldo, C. F. (2022). Parallel OpenMP and OpenACC Porous Media Simulation. *The Journal of Supercomputing*.
- Lucca, N. (2022). Avaliação de estratégias de paralelismo em simulação de meios porosos. Master's thesis, Dissertação (Mestrado Profissional em Engenharia de Software) – Universidade Federal do Pampa, Campus Alegrete, Alegrete/RS.
- Lucca, N., Schepke, C., and Tremarin, G. D. (2023). Parallel Directives Evaluation in Porous Media Application: A Case Study. In *2023 31th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, volume 1.