

Comparação de desempenho entre execuções em CPU e em GPU de implementação do método CSEM 3D

Rômulo T. Lima^{1,2}, Mateus F. Lima de Souza^{1,3}, Ramoni Z. S. Azevedo¹, Antônio Tadeu A. Gomes¹, Roberto P. Souto¹, Tiziano Labruzzo^{1,4}, Andrea Zerilli^{1,4}

¹ Laboratório Nacional de Computação Científica (LNCC)
Getúlio Vargas Av., 333, Quitandinha Petrópolis - RJ - Brasil

²Universidade Católica de Petrópolis (UCP)
R. Barão do Amazonas, 124 - Centro, Petrópolis - RJ - Brasil

³Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-FR)
R. Gen. Canabarro, 485 - Maracanã, Rio de Janeiro - RJ - Brasil

⁴Zlemlink Ltda
Rua Taylor 39, sala 805, Rio de Janeiro-RJ - Brasil

{romulotl, facanha, razaseaz, atagomes, rpsouto, tiziano}@lncc.br

Abstract. *A CPU vs. GPU performance comparison was performed for an implementation of the 3D CSEM method, which is a geophysical technique used to locate oil reserves on the seabed. A routine from the PETSc library dominates the total execution time, and its call can be executed on either the CPU or the GPU.*

Resumo. *Foi realizada uma comparação de desempenho CPU vs. GPU de implementação do método CSEM 3D, que é uma técnica geofísica usada para localizar reservas de petróleo no fundo do mar. Uma rotina da biblioteca PETSc domina o tempo total de execução, sendo possível a sua chamada ser executada em CPU ou em GPU.*

1. Introdução

Neste trabalho é apresentada uma comparação de desempenho CPU vs. GPU realizada no supercomputador Santos Dumont, de uma implementação do método Controlled-Source Electromagnetic (CSEM) [Zerilli et al. 2014], que consiste em uma técnica geofísica usada para explorar o subsolo em busca de reservas de petróleo e gás, e de outros recursos minerais. O problema associado ao método CSEM é representado pela forma integral da equação de Maxwell, cuja resolução numérica é obtida por diferenças finitas, que resulta em um sistema linear de equações, resolvido por um *solver* iterativo paralelo (**BiCGstab**) da biblioteca PETSc¹ [Balay et al. 1997], que foi desenvolvido a fim de extrair desempenho utilizando processamento heterogêneo, em CPU multi-core e GPU.

2. Metodologia

Foram realizadas execuções paralelas no supercomputador Santos Dumont (SDumont), localizado no Laboratório Nacional de Computação Científica (LNCC), o qual possui nós

¹<https://petsc.org/>

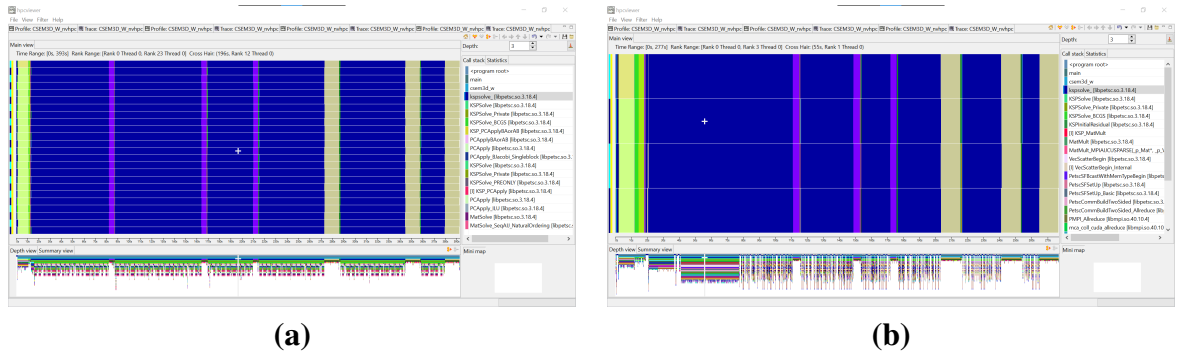


Figura 1. Perfis de desempenho paralelo do código CSEM 3D obtidos com o pacote HPCToolkit: com 24 processos MPI em CPU (a) e com 4 processos MPI em GPU (b).

contendo dois (2) sockets de CPU multi-core Intel® Xeon® Gold 6252 de 24 núcleos, totalizando portanto 48 núcleos computacionais, além de quatro (4) GPUs NVIDIA Volta V100. O pacote utilizado para coletar métricas de desempenho foi o HPCToolkit [Adhianto et al. 2010], que consiste em um conjunto de ferramentas projetado para auxiliar no enendimento e otimização no desempenho de aplicações em sistemas de HPC.

3. Resultados

Os resultados de desempenho apresentados na Tabela 1 foram obtidos utilizando os dados de um receptor do *dataset Jubarte*, rodando em uma malha 3D contendo aproximadamente 8,6 milhões de pontos ($nx = 273, ny = 193, nz = 164$). A diferença das versões executadas em CPU e em GPU consiste sobretudo no modo como é realizada a chamada para a rotina **ksp_solve** da biblioteca PETSc (versão 3.18.4), que pode ser utilizada em ambas arquiteturas computacionais. O tempo total de processamento em CPU foi de 393 segundos, enquanto que o tempo em GPU foi de 277 segundos, obtendo-se um ganho de desempenho de aproximadamente 1,4 vezes.

Na Figura 1 é mostrada a saída da visualização de execução paralela, respectivamente em CPU (a) e em GPU (b). Ressalta-se que em CPU foram utilizados 24 processos MPI, e que em GPU foram empregados 4 processos MPI, ou seja, um processo MPI por GPU disponível no nó do SDumont. No caso de CPU, destaca-se ainda que o desempenho paralelo com MPI utilizando todos os 48 núcleos do nó apresenta desempenho inferior comparado com o obtido utilizando 24 núcleos [de Souza et al. 2023]. A cor azul escuro é referente ao processamento da rotina **ksp_solve**, que corresponde a cerca de 82% do tempo total de execução paralela em CPU, e de 74% da execução usando GPU.

Foram também realizados comparativos de desempenho entre CPU e GPU com o

Tabela 1. Tempos e percentuais de tempo das rotinas chamadas pelo CSEM3D_W

Rotinas	CPU (s)	CPU (%)	GPU (s)	GPU (%)
Todas	393,0	100,0%	277,0	100,0%
ksp_solve	323,0	82,2%	205,5	74,2%
e_tx_b_rx	39,3	10,0%	36,3	13,1%
e_tx_e_rx	13,4	3,4%	11,9	4,3%

Tabela 2. Comparativo de desempenho CPU/GPU com *dataset* Marlim.

Arquitetura	#Nós	#MPI ranks	Tempo(s)	Ganho
CPU	4	96	1383,1	1,00
GPU	4	16	742,7	1,86
CPU	8	192	840,6	1,00
GPU	8	32	447,2	1,88

dataset **Marlim**, que possui um volume de dados consideravelmente maior que Jubarte, contendo cerca de 45,2 milhões de pontos de malha, ($n_x = 599, n_y = 608, n_z = 124$), sendo necessário utilizar um número maior de nós computacionais do SDumont. Na Tabela 2 é mostrado o desempenho em CPU e em GPU para um único receptor na frequência de 1,25Hz, sendo executado em 4 e em 8 nós computacionais. Nos dois cenários, o ganho de desempenho com GPU foi de aproximadamente 1,9 vezes com relação ao CPU multi-core. Este ganho é superior ao observado com Jubarte (1,4 vezes), devido possivelmente a maior granularidade de processamento com os dados de Marlim. Cabe ressaltar que com um número seis vezes menor de processos MPI, o desempenho com GPU foi quase duas vezes superior ao alcançado com CPU.

4. Comentários

Foi possível alcançar um bom ganho de desempenho com o uso de GPU na chamada do *solver* iterativo paralelo (**BiCGstab**) da biblioteca PETSc. Como um dos próximos passos, pretende-se estender o uso de GPU em nova estratégia de execução, que combine paralelismo de tarefas e de dados, explorando o recurso de instâncias independentes em uma única GPU, presente em algumas das arquiteturas da NVIDIA.

Agradecimentos

Os autores agradecem a Petrobras pelo apoio à pesquisa (Termo de Colaboração 0050.0121778.22.9), e ao LNCC por fornecer recursos do supercomputador SDumont.

Referências

- Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., and Tallent, N. R. (2010). HPCToolkit: tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, 22(6):685–701.
- Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). Efficient management of parallelism in object oriented numerical software libraries. In Arge, E., Bruaset, A. M., and Langtangen, H. P., editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press.
- de Souza, M., Lima, R., Souto, R. P., Gomes, A. T. A., Labruzzo, T., and Zerilli, A. (2023). Avaliação de desempenho de implementação paralela do método CSEM 3D no supercomputador Santos Dumont. In *Anais da VIII Escola Regional de Alto Desempenho do Rio de Janeiro*, pages 29–31, Porto Alegre, RS, Brasil. SBC.
- Zerilli, A., Labruzzo, T., Zanzi, M., Buonora, M. P., Crepaldi, J. L., and Menezes, P. T. (2014). Broadband marine CSEM: New benefits for subsalt and around salt exploration. In *2014 SEG Annual Meeting*. OnePetro.