Avaliação de Desempenho no Supercomputador SDumont de uma Estratégia de Decomposição de Domínio usando as Funcionalidades de Mapeamento Topológico do MPI para um Método Numérico de Escoamento de Fluidos.

Stiw Herrera[1],Weber Ribeiro [1], Thiago Teixeira[1], André Carneiro[1], Frederico L. Cabral[1], Márcio R. Borges[1], Carla Osthoff[1]

[1]Laboratório Nacional de Computação Cientifica (LNCC)
Av. Getúlio Vargas, 333. Quitandinha - 25651-075 - Petrópolis - RJ - Brasil

{stiw,webergdr,tteixeira,andrerc,fcabral,mrborges,osthoff}@lncc.br

Abstract. *Oil and gas simulations need new high-performance computing techniques to deal with the large amount of data allocation and the high computational cost that we obtain from the numerical method. The domain decomposition technique (domain division technique) was applied to a three-dimensional oil reservoir, where the MPI (Message Passing Interface) allowed the creation of a uni, bi and three-dimensional topology, where a subdivision of a reservoir could be solved in each MPI process created. A performance study was developed with these domain decomposition strategies in 20 computational nodes of the SDumont Supercomputer, using a Cascade Lake architecture.*

Resumo. *Simulações da área de óleo e gás precisam de novas técnicas de computação de alto desempenho para poder lidar com a grande quantidade de alocação de dados e com o alto custo computacional que obtemos do método numérico. Assim, a técnica de decomposição de domínio (divisão de domínio) foi aplicada num reservatório de petróleo tridimensional, onde o MPI (Message Passing Interface) permitiu a criação de uma topologia unidimensional, bidimensional e tridimensional, de tal forma que uma subdivisão de um reservatório possa ser resolvida em cada processo MPI criado. Foi realizado um estudo de desempenho com essas estratégias de decomposição de domínio em 20 nós computacionais no supercomputador SDumont, utilizando a arquitetura Cascade Lake.*

1. Introdução

A simulação numérica de reservatórios de petróleo consiste na elaboração de modelos matemáticos, representativos da física típica de escoamentos em meios porosos, cujas as soluções são aproximadas por métodos numéricos apropriados [Correa and Borges 2013, Murad et al. 2013]. Seu objetivo é obter um comportamento aproximado da realidade para a realização de previsões do processo de produção. As heterogeneidades, presentes nas propriedades das rochas reservatório (porosidade, permeabilidade, módulo de Young, etc.), ocorrem em todas as escalas de comprimento, desde a escala do poro até a escala de campo, que se estende por quilômetros. Tais heterogeneidades exercem marcante efeito sobre o padrão de escoamento. Portanto, em simulações típicas, precisamos discretizar domínios gigantescos (km^3) em malhas suficientemente refinadas para representar tais heterogeneidades (m^3), o que dá origem a problemas computacionais de grande porte que exigem computação de alto-desempenho para que os mesmos sejam computados em

tempo razoável. Os supercomputadores da atualidade possuem grande capacidade de memória e altíssima velocidade de processamento. Assim, o desenvolvimento de novos métodos numéricos precisam do acompanhamento de estratégias de computação de alto desempenho [Straatsma et al. 2017] para poder tirar proveito aos supercomputadores. Desta forma, diversos trabalhos estão sendo desenvolvidos com a finalidade de criar técnicas com grande escalabilidade, tanto para arquiteturas de memoria compartilhada, distribuída ou híbrida. Dentre os trabalhos com a estratégia de decomposição de domínio [Bjørstad et al. 2018] podemos mencionar: O trabalho de [Palin 2007], que foi apresentado técnicas de decomposição de domínio e processamento paralelo com troca de mensagens usando o MPI para um modelo de fenômeno físico. O trabalho de [Lima 2017] apresenta diversos métodos e exemplos de divisão de domínio para diversos tipos de malhas em arquiteturas de memoria distribuída. No artigo [Carneiro et al. 2018a] foi investigado o desempenho das operações de E / S coletivas no Supercomputados Sdumont mostrando como as diversas implementações do MPI apresentam um gargalo em relação às operações coletivas isso a medida em que aumenta o número de nós computacionais.

A contribuição deste trabalho está em apresentar um estudo de desempenho sobre a técnica de Decomposição de Domínio aplicada a uma simulação real de óleo e gás no Supercomputador Sdumont. Os testes realizados neste trabalho demonstraram que o algoritmo de divisão de domínio implementado permite minimizar a comunicação entre os processos, de forma que os gastos com a rotinas MPI não são o gargalo ao executar a malha em um ambiente de sistemas distribuídos. Os testes demonstraram que a medida em que se aumenta a malha o gargalo passa a ser o tempo gasto por uma função que precisa identificar a localização dos componentes da malha, e outra função com o acesso aos dados que utiliza o método numérico.

Este artigo é organizado da seguinte forma: A seção 2 descreve Abordagens da Decomposição de Domínio com o MPI; A seção 3 descreve os Testes realizados e por fim, a seção 4 faz a conclusão e propõe os trabalhos futuros.

2. Abordagens da Decomposição de Domínio com o MPI

A técnica de decomposição de domínio adotada nesta metodologia é chamada de decomposição em blocos como pode ser visto no artigo [Parashar and Yotov 1998]. Esta estratégia nos ajuda na implementação pois nosso domínio de simulação é um grande paralelepípedo. Essa técnica nos possibilita realizar diferente cortes transversais verticais e horizontais como pode ser visto em [Winkelmann et al. 1999]. A aplicabilidade e confiabilidade desta estratégia foram verificados em artigos como [Lima 2017].

A cada dia novas técnicas de computação de alto desempenho vem surgindo, essas novas técnicas tentam colocar na prática portabilidade, performance e escalabilidade para aplicações científicas [Straatsma et al. 2017]. Na engenharia de reservatórios [Chen 2007] é necessário levar em consideração os aspectos já mencionados, já que as escalas dos reservatórios são de quilômetros e os tempos medidos em anos [Tuane 2012]. Com modelos matemáticos já consolidados e muitos outros métodos numéricos novos surgindo cada dia como [Murad et al. 2013]. Temos a necessidade de explorar técnicas de Computação de Alto Desempenho como os apresentados em [Malyshkin 2017]. Assim, foi incorporado em um simulador de óleo e gás a técnica de decomposição de domínio. Esta técnica consiste basicamente em dividir um domínio em vários subdomínios onde cada subdomínio pode ser computado de forma independente, preocupando-se apenas com uma troca de mensagens dos elementos vizinhos presentes na borda de cada subdomínio. Desta forma, esta estratégia foi implementada em um sistema de coordenadas,

com a criação de uma topologia virtual de processos. Fora analisado o estêncil do método numérico com o intuito de identificar as respetivas informações que devem ser trocadas entre os processos. Foi necessário identificar os as células vizinhas de cada processo e finalmente toda a parte da implementação foi realizada utilizado funcionalidades do MPI.

3. Testes realizados

Toda a análise de desempenho gerada nesse trabalho foi executada no Supercomputador SDumont utilizando de 1 a 20 nós computacionais de arquitetura Cascade Lake com as seguintes configurações: Processadores Intel(R) Xeon(R) Gold 6252 CPU @ 2.10GHz. Cada nó computacional possui 2 processadores com 24 cores físicos cada, totalizando assim 48 cores físicos por nó. Os processadores se comunicam utilizando um canal de comunicação INTEL UPI. Cada core possui 32KB em L1I e 32kb em L1D, cada core possui 1MB de L2 I+D e além disso cada chip possui 35,75MB de L3 I+D. A função Hiperthreading foi desabilitada. Foram realizadas execuções de 1 processo MPI/core físico até 960 processos MPI/core físicos para gerar o perfil de desempenho. Toda e qualquer escrita em disco do código foi desligada para a coleta dos resultados.

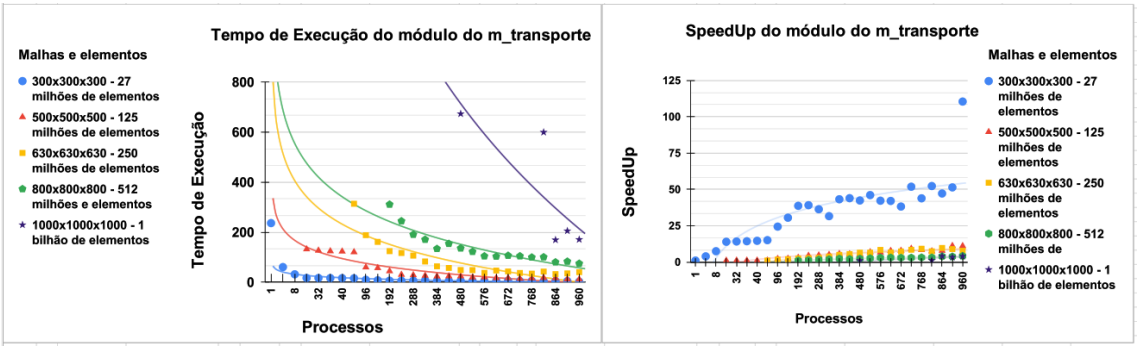

Figura 1. Gráficos do tempo de execução e do Speedup do módulo que realiza o cálculo do transporte e a comunicação entre processos, com uma execução de até 960 processos utilizando até 20 nós computacionais de arquitetura Cascade Lake do SDumont

A Figura 1 apresenta os resultados de vários testes executados em arquitetura Cascade Lake do Supercomputador SDumont. O gráfico da esquerda mostra o tempo de execução e o da direita mostra o speedup. Os resultados apresentam ganho de desempenho com esta estratégia de Decomposição de domínio, mas fica visível que quanto maior a malha, pior é o ganho de desempenho quando comparado a malhas menores. Por esse motivo realizamos alguns testes de comunicação para averiguar o possível problema.

A Figura 2 ilustra o consumo de tempo das funções de troca de mensagens do MPI durante a execução do código. O gráfico representa a execução em malhas 100x100x100, 300x300x300 e 500x500x50, utilizando 192 processos em 4 nós computacionais. Mesmo que a malha 100x100x100 apresente um grande consumo de tempo com funções MPI, fica visível que o motivo é a utilização de uma malha pequena para a grande quantidade de recursos computacionais disponíveis, causando esse desbalanceamento. Porém, ao aumentar o tamanho destas malhas, foi identificado que os gastos com as rotinas MPI não são o gargalo ao executar a malha em um ambiente de sistemas distribuídos.

A Figura 3 ilustra a comparação do *Hotspot* entre uma malha de 300x300x300 vs 100x100x100. É possível observar que, a função **sub_fvizinhos**, que identifica as células

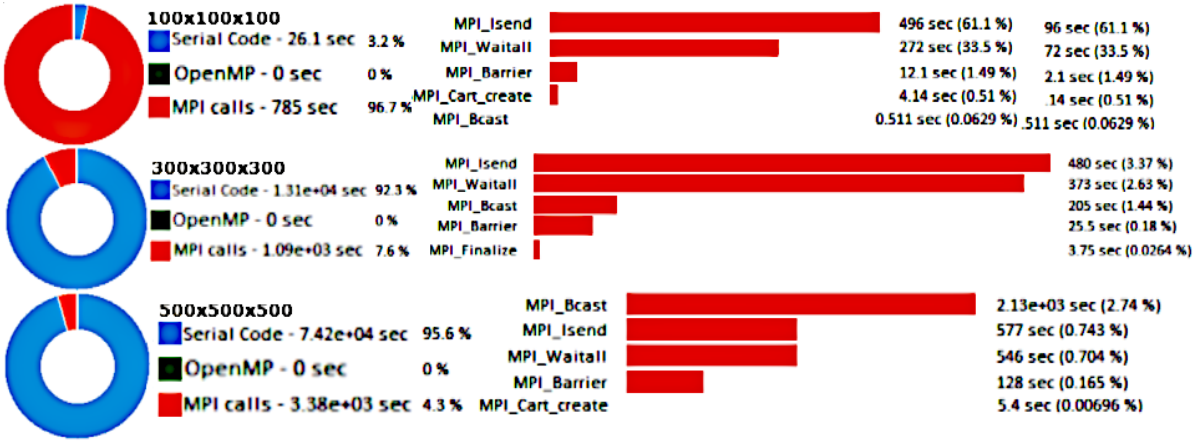

Figura 2. Histograma de comunicação em execuções com 192 processos em 4 nós utilizando malhas 100x100x100, 300x300x300 e 500x500x500 respectivamente.

Top Hotspots

Function	Module	CPU Time
sub_fvizinhos	Simulador.exec	51677.194s - 59.040s = 51618.154s
m_transportesub_transporte_mp_upwind_	Simulador.exec	782.208s - 2.482s = 779.727s
pthread_spin_lock	libpthread.so.0	501.818s - 1.719s = 500.099s
__vdso_gettimeofday	[vdso]	410.846s - 1.499s = 409.347s
m_transportesub_transporte_mp_f_upwind_	Simulador.exec	176.671s - 0.811s = 175.860s
[Others]		919.683s - 20.539s = 899.144s

Figura 3. Comparação de Hotspots para o simulador de óleo é gás usando malhas de 300x300x300 vs 100x100x100

vizinhas de cada célula, consome 51.677 segundos de um total de 54.468 segundos representando aproximadamente 94% do tempo total de execução para a malha maior e de aproximadamente 68% para malha menor, também pode ser visto que a função **upwind**, que realiza os cálculos do transporte de fluidos, teve um grande aumento no tempo de execução quando comparada a malha menor, pulando de 2.482 segundos para 782.208 segundos gerando um tempo de execução ainda maior. Estes testes demonstraram que a medida em que se aumenta a malha o gargalo passa a ser o tempo gasto com o acesso aos dados que contém a localização dos componentes da malha e no cálculo do transporte dos fluídos. Os resultados foram obtidos com o Vtune do intel parallel Studio. Também foi analisado com o Vtune com a opção *memory-access* podendo identificar que conforme aumenta a malha aumentam os acessos aos níveis de hierarquia de memoria como L1,L2,L3 para estas duas malhas, onde quanto maior a malha, maior é frequência de acesso a memória RAM ao escalonar nos níveis L2 e L3 durante a execução.

4. Conclusão e trabalhos futuros

Os experimentos apresentados neste trabalho comprovam a eficiência do método de divisão de domínio implementado. Tendo em vista os resultados apresentados, podemos concluir que as sub-rotinas chamadas de **sub_fvizinhos** e **upwind** consomem mais tempo quando são comparadas com as outras rotinas e o tempo destas funções crescem conforme o aumento do tamanho da malha, pois estas funções são encarregadas de, encontrar os vizinhos imediatos para cada elemento de volume e calcular o transporte de fluídos, respectivamente. Estas funções estão impactando diretamente no desempenho do código. Desta forma, como trabalhos futuros, pretendemos criar estrategias de localidade e acesso ao

cache. Também pretendemos avaliar o desempenho para um maior número de nós computacionais com malhas maiores para avaliar o impacto do desempenho das operações coletivas do MPI conforme o trabalho [Carneiro et al. 2018b]. Além disso, já iniciamos pesquisas em novas estratégias de E/S buscando uma melhor performance na escrita em disco, visualização e criação de *checkpoint* durante as simulações.

Referências

Bjørstad, P. E., Brenner, S. C., Halpern, L., Kim, H. H., Kornhuber, R., Rahman, T., and Widlund, O. B. (2018). *Domain Decomposition Methods in Science and Engineering XXIV*. Springer.

Carneiro, A. R., Bez, J. L., Boito, F. Z., Fagundes, B. A., Osthoff, C., and Navaux, P. O. A. (2018a). Collective i/o performance on the santos dumont supercomputer. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 45–52. IEEE.

Carneiro, A. R., Bez, J. L., Boito, F. Z., Fagundes, B. A., Osthoff, C., and Navaux, P. O. A. (2018b). Collective i/o performance on the santos dumont supercomputer. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 45–52. IEEE.

Chen, Z. (2007). *Reservoir simulation: mathematical techniques in oil recovery*. SIAM.

Correa, M. and Borges, M. (2013). A semi-discrete central scheme for scalar hyperbolic conservation laws with heterogeneous storage coefficient and its application to porous media flow. *International Journal for Numerical Methods in Fluids*, 73(3):205–224.

Lima, I. d. C. M. (2017). Simulação de reservatórios de petróleo em paralelo utilizando malhas não-estruturadas 2d e 3d.

Malyshkin, V. (2017). *Parallel Computing Technologies: 14th International Conference, PaCT 2017, Nizhny Novgorod, Russia, September 4-8, 2017, Proceedings*, volume 10421. Springer.

Murad, M. A., Borges, M., Obregón, J. A., and Correa, M. (2013). A new locally conservative numerical method for two-phase flow in heterogeneous poroelastic media. *Computers and Geotechnics*, 48:192–207.

Palin, M. F. (2007). *Técnicas de decomposição de domínio em computação paralela para simulação de campos eletromagnéticos pelo método dos elementos finitos*. PhD thesis, Universidade de São Paulo.

Parashar, M. and Yotov, I. (1998). An environment for parallel multi-block, multi-resolution reservoir simulations. In *Proceedings of the 11th International Conference on Parallel and Distributed Computing Systems (PDCS 98), Chicago, IL, International Society for Computers and their Applications (ISCA)*, pages 230–235.

Straatsma, T. P., Antypas, K. B., and Williams, T. J. (2017). *Exascale scientific applications: Scalability and performance portability*. CRC Press.

Tuane, V. L. (2012). Simulação numérica tridimensional de escoamento em reservátorios de petróleo heterogêneos. Master's thesis, LNCC/MCT, Petrópolis, RJ, Brasil.

Winkelmann, R., Häuser, J., and Williams, R. D. (1999). Strategies for parallel and numerical scalability of cfd codes. *Computer methods in applied mechanics and engineering*, 174(3-4):433–456.