Adaptação de código CSEM 3D para paralelismo de dados com escalonamento estático de tarefas

Mateus F. Lima de Souza[1,2], **Rômulo T. Lima**[1,3], **Antônio Tadeu A. Gomes**[1], **Roberto P. Souto**[1], **Tiziano Labruzzo**[1,4], **Andrea Zerilli**[1,4]

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

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

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

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

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

Abstract. *This work presents a performance comparison between two parallelism strategies of an MPI implementation of the CSEM (Controlled-Source Electromagnetic) method. The original version of the code uses task parallelism, while the modified version uses data parallelism. The experiments were conducted on nodes of the Santos Dumont supercomputer, and it was possible to verify a consistent improvement in performance when using the second strategy.*

Resumo. *Este trabalho apresenta uma comparação de desempenho entre duas estratégias de paralelismo de uma implementação MPI do método CSEM (Controlled-Source Eletromagnetic). A versão original do código utiliza paralelismo de tarefas, enquanto que a versão modificada emprega paralelismo de dados. Os experimentos foram conduzidos em nós do supercomputador Santos Dumont, e foi possível verificar uma consistente melhora no desempeho ao se utilizar a segunda estratégia.*

1. Introdução

Controlled-Source Eletromagnetic (CSEM) é um método de mapeamento geofisico que emprega um monitoramento eletromagnético através de sensores para mapear a resistência elétrica sub-aquática. Este método é utilizado em larga escala para diversas aplicações, tais como a exploração de hidrocarbonetos com tecnologia embarcada. Este trabalho teve como objetivo comparar o desempenho computacional de duas estratégias de implementação paralela MPI do código CSEM 3D [Zerilli et al. 2016]: utilizando o paralelismo original de tarefas, e também uma modificação que emprega paralelismo de dados.

Cada instanciação da aplicação gerencia um *dataset* formado por tantos arquivos quantas são as combinações transmissor-receptor-frequência. A aplicação é responsável por ler o conteúdo desses arquivos e invocar as rotinas para configurar a simulação

de imageamento da combinação transmissor-receptor-frequência correspondente. De acordo com resultados anteriores de desempenho paralelo com MPI do código CSEM 3D, observou-se um baixo aproveitamento da memória disponível nos recursos computacionais utilizados [de Souza et al. 2023]. Investiu-se então na implementação de em uma versão alternativa capaz de explorar paralelismo de dados no processamento das múltiplas combinações transmissor-receptor-frequência envolvidas na simulação de um imageamento sísmico. A seguir, é apresentada a metodologia de implementação e, na sequência, os resultados alcançados com essa estratégia.

2. Metodologia

Na versão original da aplicação, o *dataset* é processado usando um único comunicador MPI (`MPI_COMM_WORLD`) criado na sua inicialização, e cada um dos arquivos que compõem o *dataset* é processado por vez, utilizando a totalidade de processos MPI gerados para a aplicação. Na versão de paralelismo de dados, os processos MPI gerados são particionados em grupos MPI, onde cada um é responsável pelo processamento de um subconjunto dos arquivos do *dataset*, processando um arquivo por vez.

Cada grupo MPI está associado a um comunicador MPI específico criado por intermédio da função `MPI_Comm_Split`. O número de processos MPI participantes de um grupo é configurável pelo usuário. Nesta versão foi incluído um argumento de linha de comando (–group_size) que define o tamanho (inicial) do grupo MPI a ser usado. Os grupos MPI são criados por intermédio da função `MPI_Comm_Split`, que cria novos comunicadores MPI cujo número de processos associado (o tamanho do grupo MPI) é definido por esse argumento.

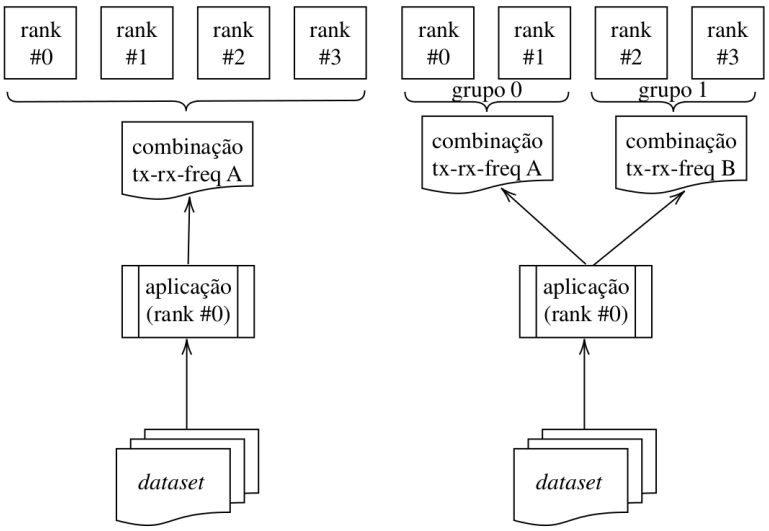
A Figura 1 ilustra como os arquivos de um *dataset* são processados pelas duas versões de aplicação descritas acima. Na figura, assume-se uma configuração de ambas as aplicações com 4 processos MPI, sendo que no caso da aplicação com paralelismo de dados, são configurados 2 grupos MPI com 2 processos MPI cada.

3. Resultados

Os nós computacionais utilizados no supercomputador Santos Dumont (SDumont) contém arquitetura de CPU multi-core Intel® Xeon® Gold 6252, possuindo 2 sockets com 24 núcleos cada, com 384GB de RAM total. Foram conduzidos dois experimentos, ambos alocando 2 nós do SDumont, com um *dataset* composto por 56 arquivos, correspondendo a um único transmissor, 14 receptores diferentes e 4 frequências distintas.

No primeiro experimento, o *dataset* foi processado usando a aplicação original do CSEM 3D, usando um *job* com 48 processos MPI no total, 24 por nó, configuração na qual é obtido um melhor desempenho, conforme mostrado em [de Souza et al. 2023]. Dessa forma, cada um dos 56 arquivos é processado por vez pela aplicação, de forma serializada. O tempo total de processamento total foi de 22.801 segundos ($\approx$ 6h20min).

No segundo experimento, o *dataset* foi processado usando a aplicação modificada para paralelismo de dados, usando um *job* com 48 processos MPI no total, 24 por nó. A distribuição dos arquivos pelos grupos MPI é feita estaticamente, isto é, antes do início do processamento. O comunicador MPI foi dividido em 8 grupos MPI com 6 processos MPI cada. Nesse caso, são distribuídos exatamente 7 arquivos para cada grupo MPI – ou seja,



(a) Versão original da aplicação do `CSEM3D_W`.

(b) Versão alternativa da aplicação do `CSEM3D_W` com paralelismo de dados.

Figura 1. Arquitetura de execução paralela do `CSEM3D_W`.

balanceamento perfeito em termos de número de combinações, mas não necessariamente em termos de tempo de processamento, devido às diferenças de tempo de convergência entre as frequências. Nesta nova configuração, o tempo total de processamento foi reduzido para 14.606 segundos ($\approx$ 4h03min).

4. Comentários

Os resultados mostram que um maior nível de paralelismo de dados trouxe benefício em termos de tempo de execução. Possivelmente um rebalanceamento logo no início (uma vez que as baixas frequências, cujo tempo de convergência para a solução é em geral mais lento, são processadas primeiro) permitiria uma redução ainda maior desse tempo de execução. Outra alternativa seria a implementação de uma versão de distribuição dinâmica dos arquivos pelos grupos MPI, o que demanda mudanças mais profundas no código.

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

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., Buonora, M. P., Menezes, P. T., Labruzzo, T., Marçal, A. J., and Silva Crepaldi, J. L. (2016). Broadband marine controlled-source electromagnetic for subsalt and around salt exploration. *Interpretation*, 4(4):T521–T531.