

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Otimização computacional do processo de inversão 2.5D de dados eletromagnéticos por uma abordagem via OpenMP

Rafael Lourenço Stanislau, Roberto P.Souto, Carla Osthoff¹

Laboratório Nacional de Computação Científica, Petrópolis, RJ

Tiziano Labruzzo, Andrea Zerilli²

Schlumberger LTDA, Rio de Janeiro, RJ

Um dos grandes problemas da geofísica é a prospecção de hidrocarbonetos em águas profundas, pois essa tarefa possui risco exploratório elevado. Além da utilização de métodos sísmicos, podem ser aplicados métodos eletromagnéticos, visto que estes permitem aos geólogos distinguirem o conteúdo dos reservatórios [2]. Para a interpretação dos dados coletados através dos métodos eletromagnéticos, faz-se necessária a aplicação de uma abordagem de inversão não-linear. A inversão é uma tarefa difícil e computacionalmente intensa, podendo demorar horas ou até mesmo dias para ser executada [1].

Por outro lado, o desenvolvimento tecnológico e científico na área de computação de alto desempenho tem gerado o aparecimento de diversas novas arquiteturas de processadores que possibilitam a execução paralela de algoritmos computacionalmente intensos, tais como as arquiteturas *multicore*, entre outros.

Este trabalho tem como objetivo reestruturar e paralelizar o algoritmo para arquiteturas *multicore* em um ambiente de computação de alto desempenho, através do modelo de programação de memória compartilhada via *OpenMP*.

A primeira etapa deste trabalho consistiu em analisar a execução do código fonte disponibilizado pela *Schlumberger*, implementado em *Fortran 90*, à procura das rotinas com maior tempo de execução, os denominados *hotspots*, compilado com *Intel Fortran 12* em modo *Debug* no *Visual Studio 2010*, na ferramenta de perfil de desempenho *Intel VTune*.

Dentre as 5 rotinas com maior tempo de execução, 3 pertencem ao modelo inverso, são elas: `COMPUTE_QNA_HESSIAN`, `DLLT0` e `COMPUTE_RES_IRHS`. Se somarmos o tempo de execução dessas 3 rotinas, elas consomem mais de 90% do tempo total de execução da aplicação.

A execução da aplicação foi realizada numa *workstation* disponibilizada pela *Schlumberger*³ que está localizada no Laboratório Nacional de Computação Científica (LNCC). Essa máquina possui dois processadores Intel Xeon E5-2687w com 8 núcleos cada, totalizando 16 núcleos, e com uma memória principal de 184 GB, contendo o *Intel Parallel*

¹rafaels,rpsouto,osthoff@lncc.br

²TLabruzzo,AZerilli@slb.com

³<https://www.slb.com>

Studio XE 2011. Os resultados apresentados foram obtidos pela média do tempo de execução em 3 execuções distintas.

O estudo de caso consiste em dados sintéticos obtidos através da ferramenta de interface gráfica SCS2D, desenvolvida pela *Zonge International*⁴, cujo domínio de entrada é especificado por um arquivo de parâmetros, possuindo 2.073 *kbytes*, sendo então considerado de pequeno porte para este tipo de análise.

No que se refere à rotina COMPUTE.QNA.HESSIAN, após a retirada da condição de corrida, encontrada durante a execução do *loop* paralelizado com *OpenMP*, foi possível obter um ganho de 15,28 vezes com a utilização de 16 *threads* e eficiência paralela de mais de 90%, mostrando uma excelente distribuição da região paralelizada entre os núcleos de processamento disponíveis.

Em substituição a rotina original DLLT0, foi implementada de uma rotina otimizada para fatoração de *Cholesky* presente na biblioteca MKL e denominada DLLT02, com a qual obteve um ganho 387 vezes na execução paralela com *OpenMP* com a utilização de 16 *threads* com uma eficiência paralela de 35%.

Em relação a rotina COMPUTE.RES.IHRS, foi realizada a paralelização da subrotina COMPUTE.JACOBIAN, gerando um ganho de desempenho de 15.50 vezes quando utilizado 16 *threads* e eficiência paralela de 96%.

Conforme os resultados apresentados, mostramos que através da reestruturação de código, da implementação de bibliotecas otimizadas, e da execução paralela em arquitetura *multicore*, foi possível reduzir de forma significativa o tempo de processamento da aplicação com relação ao código original, executado serialmente.

Vale ressaltar, que durante o processo de modificação do algoritmo original, certificou-se que todas as otimizações efetuadas no código fonte não causaram qualquer modificação na estrutura ou dados do modelo. Maiores detalhes podem ser encontrados em [3].

Referências

- [1] A. Abubakar, T.M. Habashy, V. L. Druskin, L. Knizhnerman, D. Alumbaugh. 2.5D forward and inverse modeling for interpreting low-frequency electromagnetic measurements. *Geophysics*, vol. 73, no. 4, p. F165-F177, 2008. DOI: 10.1190/1.2937466
- [2] A. Abubakar, T. M. Habashy, M. Li, and J. Liu. Inversion algorithms for large-scale geophysical electromagnetic measurements, *Inverse Problems*, volume 25, no. 12, 2009. DOI: 10.1088/0266-5611/25/12/123012
- [3] R.L. Stanislau, D.E.M Oliveira, R.P.Souto. Otimização computacional do processo de inversão 2.5D de dados eletromagnéticos em ambiente multicore. Trabalho de Conclusão de Curso do Curso de Tecnólogo em Tecnologia da Informação e da Comunicação da Faeterj/Petrópolis, 2017.

⁴<http://zonge.com/instruments-home/software/modeling-amt>