

Comparação do Custo Computacional entre Diferentes Tamanhos de Camadas de Absorção

Bruno M. O. Teixeira¹, Hermes Senger², Jonas M. Targino¹, Jaime F. de Souza²
Keith J. Roberts¹, Edson S. Gomi¹

¹Departamento de Engenharia da Computação e Sistemas Digitais –
Universidade de São Paulo (USP)

²Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
{gomi, krober, bmarizot, jonas.mendonca}@usp.br,
{senger.hermes, jaimefreire.souza}@gmail.com

Abstract. *One of the key steps in finding oil reservoirs is identifying the geology in the subsurface. An important stage of mapping the underground structures consists of simulating the propagation of acoustic waves. However, when doing so, reflexions are created in the boundaries of the domains. In order to reduce such circumstance, one may add an absorption layer to the domain. This article analyzes the computational cost of increasing the size of said layer.*

Resumo. *Uma das etapas fundamentais na busca de reservatórios de petróleo é a identificação da geologia presente na subsuperfície, de modo a tentar localizar possíveis acumulações de petróleo e gás. Uma parte importante no processo de mapeamento das estruturas no subsolo consiste em simular a propagação de ondas acústicas. Porém, ao simular propagações de onda, são produzidas reflexões indesejadas nas bordas dos domínios. Para reduzir esse efeito, podem ser adicionadas camadas de absorção aos domínios. Neste trabalho apresentamos uma análise do custo computacional do aumento do tamanho da camada de absorção.*

1. Introdução

A geofísica de exploração tem como um de seus objetivos revelar reservatórios de gás e petróleo, que muitas vezes se encontram a quilômetros abaixo da superfície do mar. Uma forma de se obter dados para estudo das estruturas geológicas é a prospecção geofísica (Figura 1), na qual uma perturbação sonora é produzida na superfície, e as ondas que refletem no subsolo são captadas por hidrofones, no caso da prospecção em áreas oceânicas. Os dados das reflexões captados formam o sismograma (Figura 1), que pode ser utilizado para identificar os reservatórios de óleo e gás.

Uma etapa do processo de identificação de reservatórios de óleo e gás é a geração de uma matriz que descreve a velocidade do som em cada ponto do domínio, chamada perfil de velocidades. O conjunto dos pontos do perfil de velocidades é chamado de *grid*. Para se estimar o perfil de velocidades, é feita uma simulação de propagação da onda acústica, que é descrita por equações diferenciais parciais (EDPs) [Louboutin et al. 2017]:

$$\begin{cases} m \frac{d^2 u(x,t)}{dt^2} - \nabla^2 u(x,t) + \eta \frac{du(x,t)}{dt} = q \text{ in } \Omega \\ u(\cdot, 0) = 0 \\ \frac{du(x,t)}{dt} |_{t=0} = 0 \end{cases} \quad (1)$$

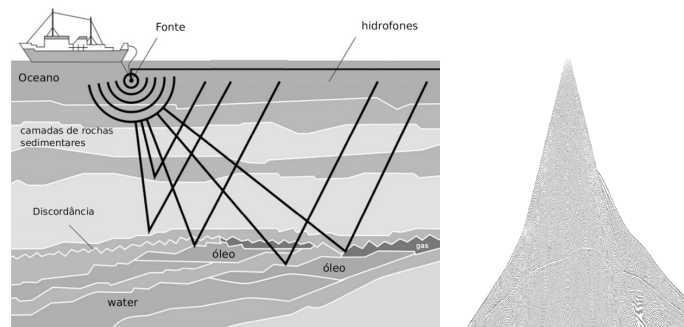


Figura 1. Prospecção em alto mar (adaptado de [Energy 2020]) e sismograma

Em (1), u é a amplitude da onda, η é o coeficiente de amortecimento, q é a fonte do sinal acústico, x é a posição espacial, t representa o tempo, e $m = \frac{1}{c^2}$, onde c é a velocidade do som no meio.

Uma forma de se resolver essas EDPs numericamente é utilizando o método das diferenças finitas [Langtangen 2017]. O Devito [Louboutin et al. 2019] é uma linguagem específica de domínio (DSL) capaz de discretizar e resolver numericamente as EDPs pelo método das diferenças finitas, gerando uma equação para calcular a intensidade u da onda em cada ponto do *grid*. Essa DSL recebe equações simbólicas, como em (2), que descrevem (1), e as convertem em código C otimizado.

$$\text{pde} = \text{model.m} * u.\text{dt}^2 - u.\text{laplace} + \text{model.damp} * u.\text{dt} \quad (2)$$

Ao se calcular a propagação da onda utilizando esse método, criam-se reflexões nas bordas dos domínios, devido às condições de fronteira, que não ocorreriam em um domínio real. Para minimizar esse efeito, é possível adicionar pontos nas bordas do domínio e criar uma camada de absorção. Para isso, o termo de dissipação η na equação (1) deve ser igual a 0 dentro do domínio, e ter um valor crescente dentro da camada de absorção. Ao se adicionar essa camada, o programa terá mais pontos a considerar no cálculo da EDP a cada passo temporal, tornando sua execução mais cara e demorada.

2. Descrição do Experimento

Nesse experimento, foram feitas simulações da propagação de ondas do tipo Ricker, com frequência de 27 Hz, no modelo de velocidades 2D BP2004 [Billette and Brandsberg-Dahl 2005]. O modelo consiste em uma matriz de dimensões (5395, 1911) pontos (Figura 2). Durante a simulação, a EDP é calculada nos pontos do grid a cada passo temporal. Na parte superior do domínio, que estende-se por 67,4 km de largura e 11,9 km de profundidade, foram posicionados de maneira uniforme 1348 receptores, que medem a amplitude da onda a cada passo temporal. O Devito simulou 12 segundos de propagação de onda, durante os quais os receptores captaram dados para formar os respectivos sismogramas.

As simulações foram executadas em duas máquinas diferentes. A primeira máquina foi uma instância da Amazon Web Services, que possui 32 GB de memória RAM, e processador Intel@Xeon@8175M, com 8 núcleos e 2 *threads* por núcleo. A segunda foi um cluster em Isambard [McIntosh-Smith et al. 2018] com processador ThunderX2 de arquitetura ARM, com 32 núcleos por soquete, 4 *threads* por núcleo e 128 GB de memória RAM. Em cada simulação, variou-se o número de pontos na camada

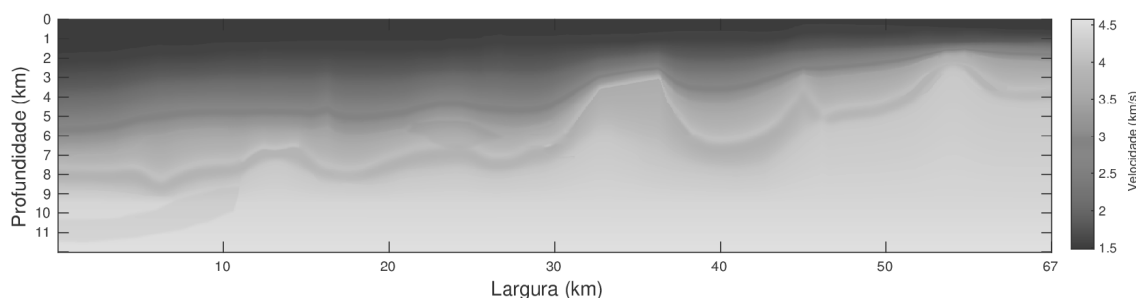


Figura 2. Modelo de velocidade BP2004 [Billette and Brandsberg-Dahl 2005]

de absorção, e mediu-se o tempo de execução do código. Foram medidos tempos de execução para modelos com camadas de absorção formadas por 0, 10, 20, 30, 100, 200, 300, e 1000 pontos. Para cada tamanho de camada de absorção, também foram feitos testes de escalabilidade, alterando-se o número de *threads* utilizadas entre 1, 2, 4, 8, e 16, no processador Intel, e entre 1, 2, 4, 8, 16, 32, 64 e 128, no processador ARM.

3. Análise dos Resultados

A partir dos resultados, é possível concluir que aumentando o número de *threads* além do número de núcleos, o ganho em tempo torna-se muito baixo. Dobrando o número de *threads* de 8 para 16, na execução no processador Intel com 0 pontos na camada de absorção, por exemplo, o tempo de execução se reduz em apenas 6%. Da mesma forma, quando o número de *threads* no processador ARM supera 32, a melhoria no desempenho não só diminui, mas fica negativa, ou seja, o tempo de execução passa a crescer conforme mais *threads* são utilizadas. Isso ocorre devido ao fato de que adicionar *threads* para executar o programa aumenta o número de acessos à memória, o que gera contenção de recursos e torna as execuções mais longas.

As simulações de propagação de onda em modelos com diferentes tamanhos de camadas de absorção mostram que, para os valores testados, tanto na máquina Intel, quanto no processador ARM, os tempos de execução aumentam conforme são adicionados mais pontos à camada de absorção (ver Figura 3). Isso acontece porque as camadas de absorção adicionam pontos nos quais a EDP terá de ser calculada a cada passo de tempo, aumentando o custo computacional. Também é possível observar um comportamento não linear no aumento do tempo de execução causado pela camada de absorção. O domínio (*grid*) tem 10,3 milhões de pontos, e quando inserimos uma camada de 1000 pontos nas laterais e abaixo, o aumentamos para 21,5 milhões de pontos, porém o tempo de execução aumenta em cerca de 3,5 vezes. Isso é explicado pelo fato de que, quando há camada de absorção, a DSL utilizada (Devito) cria código diferente para resolver a EDP, para levar em conta o termo de dissipação. Portanto, o aumento em tempo se deve não só ao maior número de pontos, mas também à maior quantidade de operações em ponto flutuante que precisam ser realizadas em modelos com camadas de absorção.

4. Conclusões

A partir dos experimentos, foi possível observar que aumentar o tamanho da camada de absorção causa um aumento no tempo de processamento de simulações de propagação de onda, tanto na máquina com processador Intel@Xeon 8175M, quanto no cluster de processadores ARM. Os gráficos na Figura 3 mostram que um número de *threads* maior do que a quantidade de núcleos do processador não melhora significativamente o desempenho, podendo até piorar. Nesses gráficos, cada ponto foi obtido calculando-se a

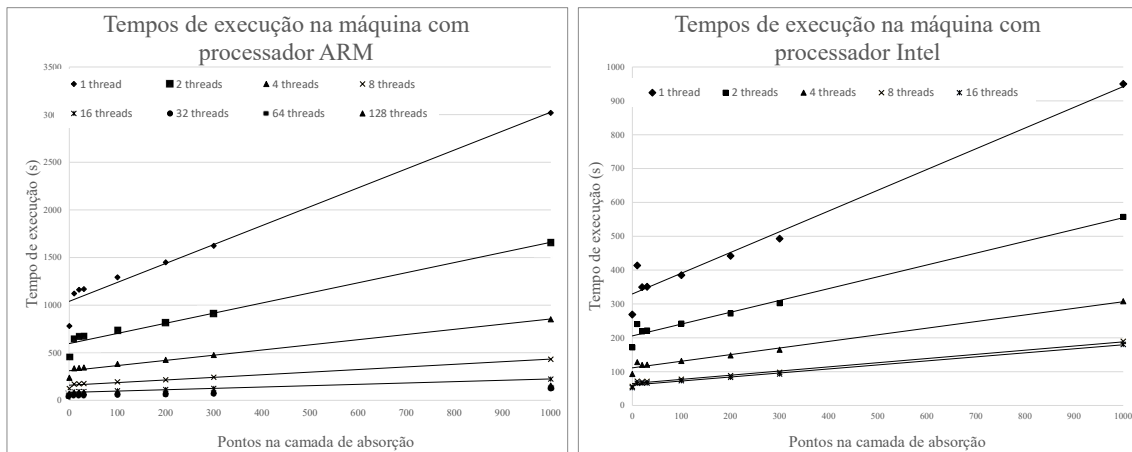


Figura 3. Tempos de execução (segundos) para cada tamanho de camada de absorção variando-se o número de cores

média de 10 execuções. Como trabalhos futuros, pretendemos fazer testes semelhantes para avaliar o efeito da camada de absorção em simulações mais complexas, como o FWI (*Full-Waveform Inversion*) [Louboutin et al. 2017]. Também seria interessante observar os efeitos que a camada de absorção tem no desempenho da simulação quando executada em GPUs.

Agradecimentos

Esta pesquisa foi desenvolvida no âmbito do projeto de P&D com registro ANP 20714-2 - Desenvolvimento de técnicas numéricas e software para problemas de inversão com aplicações em processamento sísmico (USP / Shell Brasil / ANP), patrocinado pela Shell Brasil através do compromisso com “Investimentos em Pesquisa e Desenvolvimento” com a ANP.

Referências

- Billette, F. and Brandsberg-Dahl, S. (2005). The 2004 BP velocity benchmark. In *67th Annual Internat. Mtg., EAGE, Expanded Abstracts*, page B035. EAGE.
- Energy, K. (2020). Exploration – KrisEnergy 2020. <https://krisenergy.com/company/about-oil-and-gas/exploration/>. Acesso em 3 de março de 2020.
- Langtangen, H. (2017). *Finite difference computing with PDEs : a modern software approach*. Springer, Cham, Switzerland.
- Louboutin, M., Lange, M., Luporini, F., Kukreja, N., Witte, P. A., Herrmann, F. J., Velsko, P., and Gorman, G. J. (2019). Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, 12(3):1165–1187.
- Louboutin, M., Witte, P., Lange, M., Kukreja, N., Luporini, F., Gorman, G., and Herrmann, F. J. (2017). Full-waveform inversion, part 1: Forward modeling. *The Leading Edge*, 36(12):1033–1036.
- McIntosh-Smith, S., Price, J., Deakin, T., and Poenaru, A. (2018). Comparative benchmarking of the first generation of hpc-optimised arm processors on isambard. In *Cray User Group*.