

Paralelização do Algoritmo de Migração Sísmica em Plataformas Heterogêneas

Thiago S. F. X. Teixeira¹, Wagner Meira Jr¹(Orientador), Jairo Panetta (Co-orientador)²
Banca Avaliadora: Siang W. Song³, Dorgival O. Guedes Neto¹, Renato Antônio C. Ferreira¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

²Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brasil

³Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{thiago, meira, dorgival, renato}@dcc.ufmg.br, jairo.panetta@gmail.com, song@ime.usp.br

Palavras-chave: Processamento paralelo e distribuído, processamento sísmico, GPGPU.

Abstract. *Petrobras daily uses Kirchhoff seismic migration to search for new oil and gas reserves. Kirchhoff is a CPU bound algorithm—a single run may use up to 1000 dedicated x86 cores during a month. The emergence of novel parallel computing architectures, such as GPUs, poses an opportunity and a challenge to reduce such execution times. In this thesis, we discuss the parallelization of the Kirchhoff seismic migration for an heterogeneous environment with CPUs and GPUs. We search and evaluate parallelism strategies that efficiently use the available hardware. We explore parallelism opportunities, by designing, implementing and evaluating various possible configurations. We also devise and implement a finer grain dynamic scheduling for the devices, achieving highly efficient executions. Experiments show an acceleration of up to 87 times over a single x86 core execution.*

Resumo. *A Petrobras usa diariamente a migração sísmica de Kirchhoff para procurar novas reservas de petróleo. Kirchhoff é um algoritmo limitado pela CPU—uma única execução pode utilizar 1000 núcleos x86 dedicados durante um mês. O surgimento de arquiteturas paralelas inovadoras, como as GPUs, apresenta oportunidades e desafios para reduzir tais tempos de execução. Nesta tese, discutimos a paralelização do algoritmo da migração sísmica de Kirchhoff para um ambiente heterogêneo com CPUs e GPUs. Pesquisamos e avaliamos estratégias de paralelismo que utilizam eficientemente o maquinário disponível. Exploramos as oportunidades de paralelismo projetando, implementando e avaliando várias configurações possíveis. Nós também projetamos e implementamos um escalonamento dinâmico de grão mais fino para os dispositivos, possibilitando execuções mais eficientes. Os experimentos apresentaram uma aceleração de até 87 vezes sobre a execução em um único núcleo x86.*

1. Introdução

O processamento sísmico é uma das tarefas de maior demanda computacional na indústria do petróleo. A Petrobras executa, em produção, a migração sísmica de Kirchhoff em

aglomerados dedicados de até 1000 núcleos x86. Uma execução pode demandar um mês.

A indústria constantemente avalia a utilização de novas arquiteturas para acelerar esse processamento e reduzir os custos de resfriamento, armazenamento e consumo de energia dos aglomerados. Atualmente, surgem novas arquiteturas paralelas que transformam os nós de processamento em ambientes hierárquicos e heterogêneos, onde cada nó pode ter múltiplos processadores diferentes, como os CPUs multi-core e as GPUs many-core.

Esses ambientes heterogêneos têm sido bastante pesquisados pela comunidade científica nos últimos anos, principalmente, por causa de seu melhor preço/watt e preço/flops. No entanto, é um desafio utilizar de maneira eficiente esse conjunto assimétrico de processadores quando o número de núcleos das CPUs e GPUs varia. Por exemplo, em aglomerados adquiridos em anos distintos, contendo nós distintos. Para se beneficiar de toda essa capacidade de processamento, as aplicações precisam escalonar as tarefas de maneira balanceada, considerando o melhor momento e a melhor unidade de processamento para cada caso. Isso leva a novas abordagens e estratégias para escalar tarefas e alcançar o melhor desempenho, sem perder a flexibilidade requerida pelas diferentes configurações dos aglomerados.

A tendência atual das novas arquiteturas é aumentar o desempenho acrescentando mais núcleos. Isso significa que para se beneficiar de todo esse poder computacional adicional será necessário implementar aplicações escaláveis, pois é inviável reimplementar a aplicação para cada novo aglomerado.

A paralelização eficiente e escalável da migração sísmica é um desafio pela própria complexidade da aplicação e suas várias oportunidades de paralelização possíveis, o que demanda o desenvolvimento de técnicas que explorem essas oportunidades em ambientes de processamento heterogêneos. Mais ainda, ela tem grande importância para a indústria do petróleo, uma vez que, quanto mais rápido os resultados forem obtidos, mais regiões podem ser exploradas na busca por óleo e gás. Assim, se torna evidente o interesse e o desafio do ponto de vista acadêmico de estudar e fornecer soluções para indústria sobre como explorar de maneira eficiente essas novas arquiteturas.

Resumidamente, as principais motivações para esse trabalho são: (i) disponibilidade de novas arquiteturas paralelas, como as CPUs multi-core e as GPUs programáveis, atualmente comuns nos ambientes de pesquisa e desenvolvimento em computação científica; (ii) utilização eficiente dessas novas arquiteturas, em que os desenvolvedores precisarão se preocupar com a escalabilidade e levar em conta as características de cada arquitetura. Além de lidar com o escalonamento de tarefas flexível para plataformas multi-aceleradas, multiprocessadas, multi-core e heterogêneas; e (iii) relevância da migração sísmica no processamento dos dados sísmicos, pois, além de ser muito utilizada, é um dos algoritmos de maior demanda computacional.

Portanto, o objetivo é o de estudar, implementar e avaliar oportunidades de paralelismo na migração sísmica em ambientes heterogêneos e variáveis para obter o máximo de eficiência e desempenho na utilização dos recursos computacionais disponíveis.

A principal contribuição desse trabalho foi a paralelização do algoritmo de migração sísmica de Kirchhoff para um ambiente heterogêneo distribuído composto de CPUs multi-core e GPUs. Apesar de as CPUs e as GPUs terem sido o alvo da

implementação, esta pode ser facilmente generalizada para outros dispositivos e ambientes com características semelhantes. A implementação foi feita de maneira escalável e flexível para permitir a utilização concorrente de vários processadores e múltiplos aceleradores em múltiplas configurações.

Três dimensões de paralelismo foram desenvolvidas para atingir os objetivos mencionados anteriormente. Essas três dimensões utilizaram assincronia, paralelismo de dados e paralelismo de tarefas para alcançar uma significativa melhora no desempenho.

Outra contribuição importante foi a implementação de um escalonamento dinâmico de tarefas de grão mais fino entre as unidades de processamento. Essa contribuição permitiu que não houvesse desbalanceamento de carga entre as múltiplas unidades heterogêneas e que todas operassem na máxima capacidade de processamento.

Parte dos resultados do trabalho de pesquisa e desenvolvimento da migração sísmica em ambientes heterogêneos foram publicados no artigo [Panetta et al. 2009]. Esse mesmo artigo foi convidado para ser estendido para a revista *International Journal of Parallel Programming* e o texto submetido foi aceito para publicação. Esse texto contém um resumo das informações consideradas mais importantes. Para mais informações veja [Teixeira 2010].

2. Migração Sísmica

O método sísmico pode ser dividido em três etapas: aquisição, processamento e interpretação. A primeira etapa, a aquisição dos dados, utiliza uma fonte de ondas sísmicas na superfície, como uma bomba ou um canhão de ar comprimido, para periodicamente gerar ondas que são refletidas nas interfaces das camadas que formam o subsolo e depois são coletadas por receptores na superfície. Um traço é o conjunto de sinais originados por uma única emissão da fonte e coletados por um único receptor. Um traço é composto por valores de amplitude denominados amostras, coletados em intervalos de tempo constantes.

O processamento dos dados sísmicos, que corresponde à segunda etapa, tem como objetivo extrair informações do subsolo a partir dos dados adquiridos. A indústria do petróleo utiliza, internacionalmente, pacotes de software proprietários que contém centenas de módulos sísmicos. Profissionais de processamento sísmico selecionam os módulos mais apropriados para a aquisição e para a área alvo. Esse processamento demanda meses de trabalho de uma equipe especializada e seu tempo de execução é totalmente dominado pelo módulo de migração sísmica. A terceira etapa corresponde à interpretação dos dados por outros especialistas, que indicam onde perfurar.

A migração é o processo de produzir uma imagem do subsolo consistente com os dados adquiridos, através do posicionamento correto das superfícies refletoras. É um problema inverso, pois produz os parâmetros do modelo a partir de dados observados. Como em muitos outros problemas inversos, soluções consistentes podem requerer múltiplas execuções. A migração sísmica de Kirchhoff [Yilmaz 1987] é um dos métodos de migração mais utilizados no processamento dos dados.

O algoritmo da migração sísmica calcula as contribuições de cada traço de entrada previamente filtrado para os traços de saída contidos em uma abertura definida pelo usuário. A migração possui duas etapas que dominam o seu tempo de execução: **filtragem**

dos traços de entrada e o **laço de migração**. Essas duas etapas podem ser sobrepostas, pois enquanto um traço de entrada está sendo migrado no laço de migração, um outro traço de entrada pode ser filtrado.

3. Paralelização da Migração Sísmica

Para aumentar a cooperação entre os dispositivos e utilizar todos os recursos computacionais de um ambiente heterogêneo, implementamos três dimensões de paralelismo, que podem se adaptar a diferentes plataformas pela simples alteração de parâmetros de execução. Nós utilizamos o Anthill [Ferreira et al. 2005] para implementar a migração sísmica em um ambiente distribuído. As dimensões de paralelismo foram:

1. *Paralelismo pelo dado de saída*: ocorre pela execução simultânea do laço de migração para um único traço de entrada e múltiplos traços de saída. Como a computação de um traço de saída é independente de qualquer outro traço de saída, este processamento é embaraçosamente paralelo e pode ter diferentes abordagens de acordo com o dispositivo usado (i.e., CPU ou GPU);
2. *Paralelismo de tarefas*: as duas tarefas que dominam o tempo de execução (filtragem e laço de migração) podem ser sobrepostas, utilizando a GPU para o laço de migração e a CPU para a filtragem. No Anthill, cada uma dessas etapas é um filtro. Como cada filtro pode ter múltiplas cópias independentes, o paralelismo de tarefas ocorre pela execução simultânea de cópias que atuam em dados distintos;
3. *Paralelismo pelo dado de entrada*: corresponde ao processamento simultâneo de vários traços de entrada para um mesmo conjunto de traços de saída. Isto acarreta uma condição de corrida pois cada traço de saída somente pode receber a contribuição de um traço de entrada por vez. Para superar esse problema, múltiplos traços de entrada são processados simultaneamente, mas em diferentes cópias dos traços de saída e, no final, essas cópias são agregadas para gerar o resultado final.

Foi utilizada uma abordagem de escalonamento de grão mais fino, por traços de entrada, entre as unidades de processamento. Assim, o mesmo conjunto de traços de saída pode receber as contribuições de vários traços de entrada em dispositivos diferentes, que podem operar na máxima capacidade, sem deixar recursos ociosos. Essa abordagem não garante um escalonamento de traços ótimo entre os dispositivos, mas como o grão é fino, a penalidade do dispositivo mais lento processar o último traço é pequena.

A figura 1 apresenta as três dimensões de paralelismo e como elas se relacionam. A etapa *Kirchhoff* na figura corresponde ao laço de migração.

4. Experimentos

Os experimentos foram feitos com o objetivo de avaliar as três dimensões de paralelismo apresentadas anteriormente. Eles foram executados utilizando um aglomerado onde cada nó contém dois processadores Intel Xeon L5420 com 4 núcleos cada, 32 GB de memória principal e uma GPU Nvidia Tesla C1060.

Utilizamos somente um nó do aglomerado pois o processamento dos blocos de traços de saída é embaraçosamente paralelo e o speedup para múltiplos nós, cada um processando um bloco por vez, é linear [Panetta et al. 2009]. A configuração utilizada

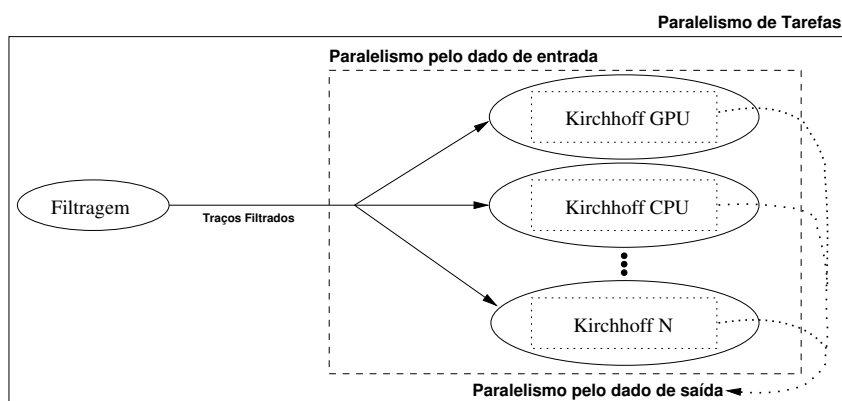


Figura 1. Três dimensões de paralelismo.

para um nó pode ser replicada para todos os outros do aglomerado e facilmente alterável (por parâmetros de execução) para outros aglomerados. Utilizamos como métrica de avaliação a taxa de amostras contribuídas por segundo. As configurações analisadas foram: sequencial na CPU (*CPU*), laço de migração paralelizado na CPU (*8 CPUs*), execução síncrona do laço de migração na GPU (*GPU sínc*), execução assíncrona do laço de migração na GPU (*GPU assínc*), execução assíncrona do laço de migração na GPU com duas instâncias para a filtragem no ambiente Anthill (*AH GPU*), execução assíncrona do laço de migração na GPU e na CPU com duas instâncias para a filtragem no ambiente Anthill (*AH GPU+CPU*). Em termos de oportunidades de paralelismo exploradas, 8 CPUs e GPU Sinc exploram paralelismo pelo dado de saída. GPU sinc, GPU assinc e AH-GPU exploram paralelismo de tarefas. Finalmente, AH-GPU e AH-GPU+CPU exploram paralelismo pelo dado de entrada.

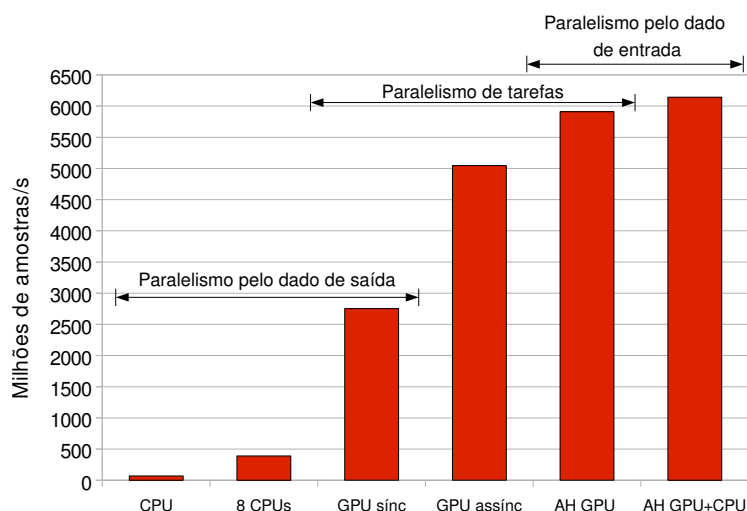


Figura 2. Análise comparativa do desempenho alcançado pelas três dimensões de paralelismo.

A figura 2 resume o desempenho alcançado em cada dimensão. A contribuição do paralelismo pelo dado de saída foi a aceleração de 5,55 vezes para 8 CPUs e de 39 vezes para a GPU (*GPU sínc*) com relação à execução sequencial na CPU. O paralelismo de tarefas ocorre tanto para o laço de migração na CPU quanto na GPU, no entanto, a contribuição foi significativa somente para o caso da GPU. A assincronia de execução entre as tarefas aumentou de 2.753 milhões (*GPU sínc*) para 5.045 milhões (*GPU assínc*) a

taxa de amostras contribuídas por segundo. O aumento no paralelismo da etapa de filtragem (*AH GPU*), através do aumento do número de instâncias no ambiente Anthill, elevou esse ganho para 5.910 milhões. Por último, a utilização conjunta entre CPU e GPU (*AH GPU+CPU*) no filtro Kirchhoff alcançou 6.143 milhões de amostras por segundo, através da utilização do paralelismo pelo dado de entrada. É importante notar que, tendo em vista as características do problema, a estratégia de paralelização pode ser replicada de forma trivial para várias máquinas que conjuguem CPUs e GPUs, utilizando uma estratégia de paralelismo de tarefas.

5. Conclusões e Trabalhos Futuros

Nesse trabalho discutimos a paralelização da migração sísmica de Kirchhoff para ambientes heterogêneos distribuídos. Para atingir o objetivo de aproveitar todos os recursos presentes, foram implementadas três dimensões de paralelismo: pelo dado de saída, de tarefas e pelo dado de entrada. A cooperação entre CPU e GPU foi priorizada ao invés de acelerar o processamento com toda a execução na GPU.

Os resultados mostraram uma redução enorme no tempo de execução com a utilização da GPU para a tarefa de maior demanda computacional. Além disso, a cooperação entre GPU e CPU, tanto na dimensão de tarefas quanto na dimensão dos dados de entrada, contribuíram ainda mais para melhorar o desempenho da aplicação. A assincronia utilizada na implementação em conjunto com a proposta pelo Anthill também contribuiu muito para o desempenho alcançado. A implementação utilizando as três dimensões propostas alcançou 6.143 milhões de amostras contribuídas por segundo, uma aceleração de até 87 vezes com relação à execução sequencial na CPU.

Como trabalhos futuros vislumbramos a aplicação dessas dimensões de paralelismo e da cooperação entre as múltiplas unidades de execução em outras aplicações. Várias aplicações de processamento sísmico se encaixam nesse perfil, como: RTM, Kirchhoff em profundidade e migração Omega-xy. Também seria interessante avaliar essa cooperação para casos em que a aceleração da GPU com relação à CPU seja menor. Assim, a CPU seria mais eficaz e novas dimensões de paralelismo podem ser exploradas.

Referências

- Ferreira, R. A., Meira Jr., W., Guedes, D., Drummond, L. M. A., Coutinho, B., Teodoro, G., Tavares, T., Araujo, R., and Ferreira, G. T. (2005). Anthill: A scalable run-time environment for data mining applications. In *SBAC-PAD '05: Proceedings of the 17th International Symposium on Computer Architecture on High Performance Computing*, pages 159–167, Washington, DC, USA. IEEE Computer Society.
- Panetta, J., Teixeira, T., de Souza, P., da Cunha, C., and et al. (2009). Accelerating kirchhoff migration by cpu and gpu cooperation. In *SBAC-PAD '09: Proceedings of the 21st International Symposium on Computer Architecture and High Performance Computing*, pages 81–90, Washington, DC, USA. IEEE Computer Society.
- Teixeira, T. S. F. X. (2010). Paralelização do algoritmo de migração sísmica em plataformas heterogêneas. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil. <http://www.dcc.ufmg.br/~thiago/ctd>.
- Yilmaz, O. (1987). *Seismic Data Processing*, volume 2 of *Investigations in Geophysics*. Society of Exploration Geophysicists, Tulsa.