Otimizando o uso do Subsistema de Memória de GPUs para Aplicações Baseadas em Estênceis*

Ricardo K. Lorenzoni[1], Matheus S. Serpa[2], Edson L. Padoin[1,2], Jairo Panetta[4]
Philippe O. A. Navaux[2], Jean-François Méhaut[3]

[1]Universidade Reg. do Noroeste do Estado do Rio G. do Sul (UNIJUI)
Ijuí – RS – Brasil

{ricardo.lorenzoni,padoin}@unijui.edu.br

[2]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970, Porto Alegre – RS – Brasil

{msserpa,navaux}@inf.ufrgs.br

[3]Universite Grenoble Alpes – Grenoble – França

jean-francois.mehaut@imag.fr

[4]Instituto Tecnológico de Aeronáutica (ITA) – S. J. Campos – Brazil

jairo.panetta@gmail.com

Abstract. *Energy and performance of parallel systems are an increasing concern for new large-scale systems. Research has been developed in response to this challenge aim the manufacture of more energy-efficient systems. In this context, this paper proposes to accelerate performance and increase the energy efficiency of stencil application by optimizing the use of the memory subsystem of GPUs. Our developed GPU-optimized algorithms for stencil applications achieve a performance speedup of up to* 2.85 *compared with the naive version. The computational results have shown that the combination of the Z-axis internalization of stencil application and the reuse of registers of architecture can achieve about* 20.24% *of energy saving and an increase of up to* 50% *in energy efficiency.*

Resumo. *O desempenho e a eficiência energética de sistemas paralelos são uma preocupação crescente para sistemas de larga escala. Pesquisas tem sido desenvolvidas em resposta a este problema focando na obtenção de sistemas com melhor eficiência energética. Neste contexto, este trabalho tem como objetivo melhorar o desempenho e a eficiência energética de aplicações baseadas em estênceis, pela otimização do uso do subsistema de memória de placas GPUs. Os resultados mostram, que o ganho de desempenho utilizando as otimizações propostas são de até* $2,85$ *vezes comparados à versão original. A otimização que combina a internalização do eixo Z com o reuso de registradores resulta em uma redução de até* $20,24\%$ *no consumo energético e um aumento de até* 50% *na eficiência energética.*

*Trabalho parcialmente apoiado por CNPq, CAPES, FAPERGS, Intel Corporation e FINEP. Pesquisa realizada no contexto do Laboratório Internacional Associado LICIA e tem recebido recursos do programa EU H2020 e do MCTI/RNP-Brasil sob o projeto HPC4E de número 689772.

1. Introdução

Simulações numéricas são utilizadas para a predição do comportamento de diversos fenômenos, sendo aplicadas em diversas áreas tais como a física, geologia, química, entre outras. A precisão e a acurácia dos métodos numéricos utilizados estão associadas aos recursos computacionais disponíveis. Um subconjunto dessas simulações utiliza estênceis, sendo que o tempo de suas computações é elevado até mesmo em supercomputadores [de la Cruz and Araya-Polo 2011].

O relatório DARPA [Bergman et al. 2008] recomenda um limite de 20MW para a criação de sistemas *Exascale*. Devido a esta recomendação, atualmente um dos desafios da Computação de Alto Desempenho (CAD) é melhorar a eficiência energética dos dispositivos. Neste sentido, reduzir o tempo total de execução de aplicações é uma forma viável de reduzir o consumo de energia, visto que energia é economizada quando recursos de *hardware* são utilizados por menos tempo. Buscando reduzir o tempo de execução, bem como reduzir o consumo de energia das aplicações, portar as aplicações científicas para execução em GPUs, tem sido uma estratégia amplamente adotada.

A computação de estênceis pode ser portada para GPUs com significativa melhoria de desempenho quando comparada com implementações realizadas em CPUs [Maruyama and Aoki 2014]. Além disto, por ser tipicamente *memory-bound*, a computação de estênceis pode beneficiar-se do uso dos diferentes níveis de memória das GPUs. Neste contexto, o objetivo deste trabalho é analisar o desempenho e a eficiência energética de uma aplicação de propagação de ondas baseada em estênceis, executada em uma arquitetura composta por uma GPU. Desta forma, será possível analisar como o uso de diferentes memórias da arquitetura impactam no desempenho dessa aplicação.

2. Trabalhos Relacionados

Diversos trabalhos têm sido desenvolvidos sobre a otimização de aplicações de propagação de ondas. Em [Nasciutti and Panetta 2016], os autores aplicam otimizações na computação de estênceis 3D e analisam o desempenho de GPUs focando no uso adequado da hierarquia de memória e concluem que a codificação mais indicada é baseada na combinação do uso do cache somente leitura, internalização do laço em Z e o reuso de registradores.

Em [Maruyama and Aoki 2014], os autores utilizaram a memória compartilhada para melhorar a localidade dos dados e a especialização de *warps* para obter uma maior taxa de saída de instruções, obtendo aproximadamente 80% do valor do modelo *roofline*.

Em seu trabalho, [Hamilton et al. 2015] investiga a performance de computação de estênceis em GPUs variando o tamanho e a forma dos estênceis. Os autores apontam que a movimentação de dados é o gargalo deste tipo de aplicações, estênceis compactos são mais eficientes utilizando a cache somente leitura, que estênceis de braço requerem uma porção significativa de banda da memória global para obter performance similar a dos estênceis compactos com quantidades de pontos similares.

Diferente destes trabalhos que analisam e otimizam o desempenho de aplicações em GPUs, e analisam o desempenho de aplicações de estênceis com diferentes formatos, o foco do presente artigo é analisar o desempenho, o consumo de energia, a demanda de potência e a eficiência energética de uma aplicação real de propagação de onda. Dando enfoque na eficiência energética, dada a importância da melhoria da eficiência energética de dispositivos na

atualidade.

3. Metodologia Experimental

Para a realização dos testes de desempenho e eficiência energética de uma aplicação estênceis, uma GPU da arquitetura Kepler foi utilziada. A GPU NVIDIA K80 possui 2496 CUDA cores, sendo que, cada *Streaming Multiprocessor (SMX)* possui uma memória *on-chip* configurável que pode ser configurada como 48/32/16 KB de memória compartilhada e 16/32/48 KB de cache L1. Ela também possui uma memória rápida somente leitura de 48 KB e uma memória cache L2 compartilhada. A Tabela 3 apresenta o ambiente utilizado.

Sistema	Parâmetro	Valor
Kepler	Dispositivo	Tesla K80
	CUDA Cores	2496 (13 SMXs $\times$ 192 SPs/SMX)
	Registradores	13 $\times$ 256 KByte
	Caches	13 $\times$ 64 KByte L1 / *compartilhada*, 1280 KByte L2
		13 $\times$ 48 KByte *texture (read-only)*
	Memória global	12 GByte GDDR5

Tabela 1. Configuração da GPU K80

Dados de desempenho e eficiência energética das aplicações de estêncil foram coletados. Para medir a potência média utilizamos a *NVIDIA Management Library (NVML)*. Para calcular a eficiência energética, utilizamos desempenho pela potência média. Os valores apresentados neste trabalho são os resultados médios de 30 execuções.

4. Computação de Estênceis em GPUs

Esta seção apresenta uma análise da computação de estênceis em diferentes níveis da arquitetura de memória da GPU apresentada na ultima seção.

Diferentes tamanhos de estêncil foram utilizados em nossa pesquisa com aplicações de estêncil. Contudo, neste artigo apresentamos os resultados obtidos com estênceis de 13, 19, 25 e 31 pontos. Portanto, para facilitar a análise, separamos os resultados obtidos em desempenho e eficiência energética de 3 versões da aplicação i) BASE, ii) READ.ONLY e iii)SHARED) aplicando diferentes otimizações.

4.1. Análise de Desempenho

Primeiramente analisamos o desempenho obtido com as diversas versões da aplicação. A Figura 1 compara o desempenho de cada uma das aplicações: (i) realizando uma computação normal com estênceis (BASE); (ii) utilizando a cache somente leitura (READ.ONLY); e (iii) utilizando a memória compartilhada. Ela também apresenta uma avaliação dos ganhos obtidos quando as otimizações são aplicadas.

O desempenho da aplicação de estênceis foi aumentado em $2,09$, $2,77$ e $2,66$ vezes sobre a BASE (linha preta) quando a otimização READ.ONLY.INT.Z.REGISTERS (linha bordô) foi aplicada nos estênceis com 13, 19 e 25 pontos respectivamente. O maior ganho ocorreu na execução sobre o estêncil de 31 pontos quando a otimização SHARED.INT.Z.REGISTERS (linha

cinza) foi aplicada. Ela aumentou em até $2,85$ vezes o desempenho se comparada com a versão BASE.

As otimizações READ.ONLY.INT.Z.REGISTERS (linha bordô) e SHARED.INT.Z.REGISTERS (linha cinza) obtiveram os maiores desempenhos em todos os testes com todos os tamanhos de estênceis. Para o estêncil de 31 pontos, READ.ONLY.INT.Z.REGISTERS e SHARED.INT.Z.REGISTERS atingiram $327,41$ e $326,91$ GFlops respectivamente.

As aplicações READ.ONLY (linha vermelha), BASE (linha preta) e BASE.INT.Z (linha azul), obtiveram desempenho linear para todos os tamanhos de estêncil e todos os tamanhos de matriz utilizados em nossos testes. A otimização READ.ONLY obteve desempenho similar a obtida pelas otimizações SHARED.INT.Z.REGISTERS e READ.ONLY.INT.Z.REGISTERS na execução com o estêncil de 13 pontos, tendo desempenhos cada vez melhores de acordo com o aumento do tamanho do problema. Porém, a medida que o tamanho do estêncil aumenta, esta otimização perde desempenho, apresentando performance similar a obtida pela otimização BASE no estêncil de 31 pontos com 2048 pontos no eixo-X.

4.2. Avaliação da Eficiência Energética

A eficiência energética é uma restrição na viabilidade da construção de novos sistemas de CAD. Vários estudos no estado da arte avaliam abordagens de *software* e *hardware* para melhorar a eficiência energética. Além destas abordagens, nossa pesquisa visa compreender como o uso da memória afeta a eficiência energética.

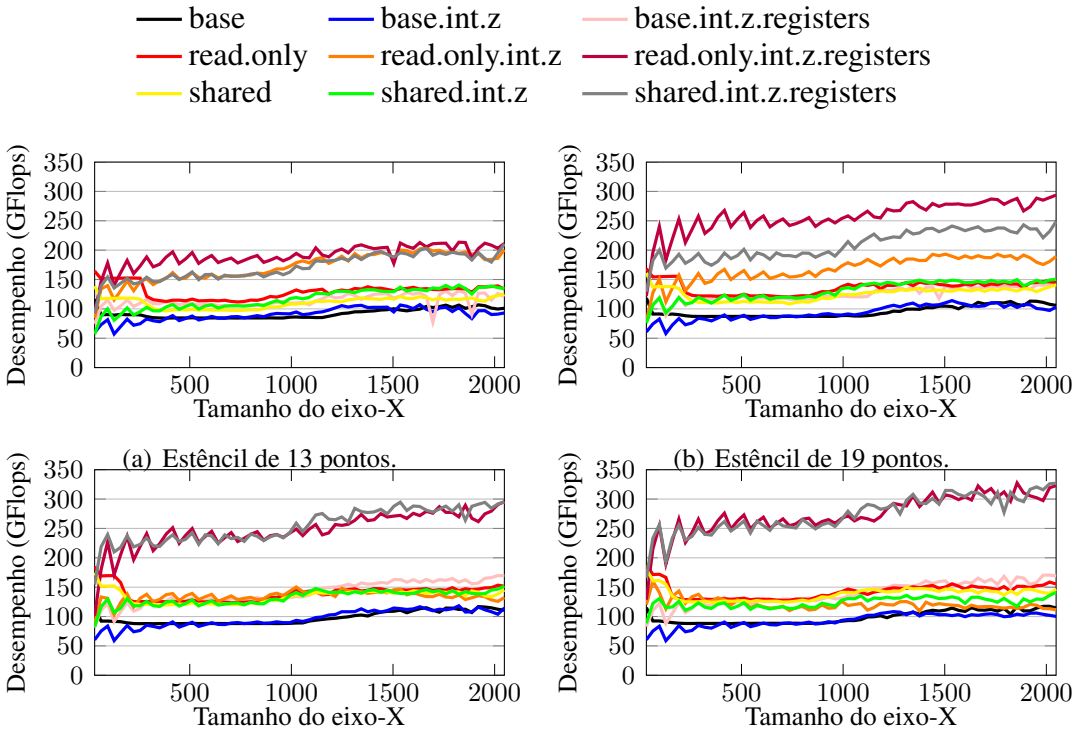
As otimizações READ.ONLY.INT.Z.REGISTERS (linha bordô) e SHARED.INT.Z.REGISTERS (linha cinza) aumentaram a desempenho para todos os tamanhos de estêncil utilizados (Figura 1). Eles também apresentam um aumento na eficiência energética durante a execução das aplicações. Desta forma, estas otimizações obtiveram a maior eficiência energética (Figura 2).

Aplicando READ.ONLY.INT.Z.REGISTERS e SHARED.INT.Z.REGISTERS no estêncil de 13 pontos com eixo-X de 2048 pontos, a eficiência energética foi de 3 e $3,3$ GFlops/W respectivamente. Obtendo um ganho de 50% e 39% quando o tamanho do estêncil utilizado é de 31 pontos com 2048 pontos no eixo-X, obtendo $4,5$ e $4,6$ GFlops/W.

A otimização READ.ONLY.INT.Z (linha laranja) obteve uma eficiência similar as das aplicações READ.ONLY.INT.Z.REGISTERS e SHARED.INT.Z.REGISTERS no estêncil de 13 pontos. Porém, quando o tamanho do estêncil aumenta, a eficiência reduz, obtendo uma eficiência energética inferior a obtida pela otimização BASE para o estêncil de 31 pontos com eixo-X de 2048 pontos.

A eficiência energética reduz quando o tamanho do estêncil aumenta. Isto pode estar relacionado a uma sobrecarga na unidade de controle da cache somente leitura, forçando interrupções na execução da aplicação enquanto aguarda pelo carregamento de dados. Então reduzindo a eficiência energética e a desempenho (Figuras 2 1).

A otimização SHARED.INT.Z apresenta uma eficiência quase linear, sendo independente do tamanho do eixo-X. Esta aparente estabilidade da eficiência energética está relacionada ao aumento de potência necessária para a execução, bem como o ganho de desempenho obtido por esta otimização.



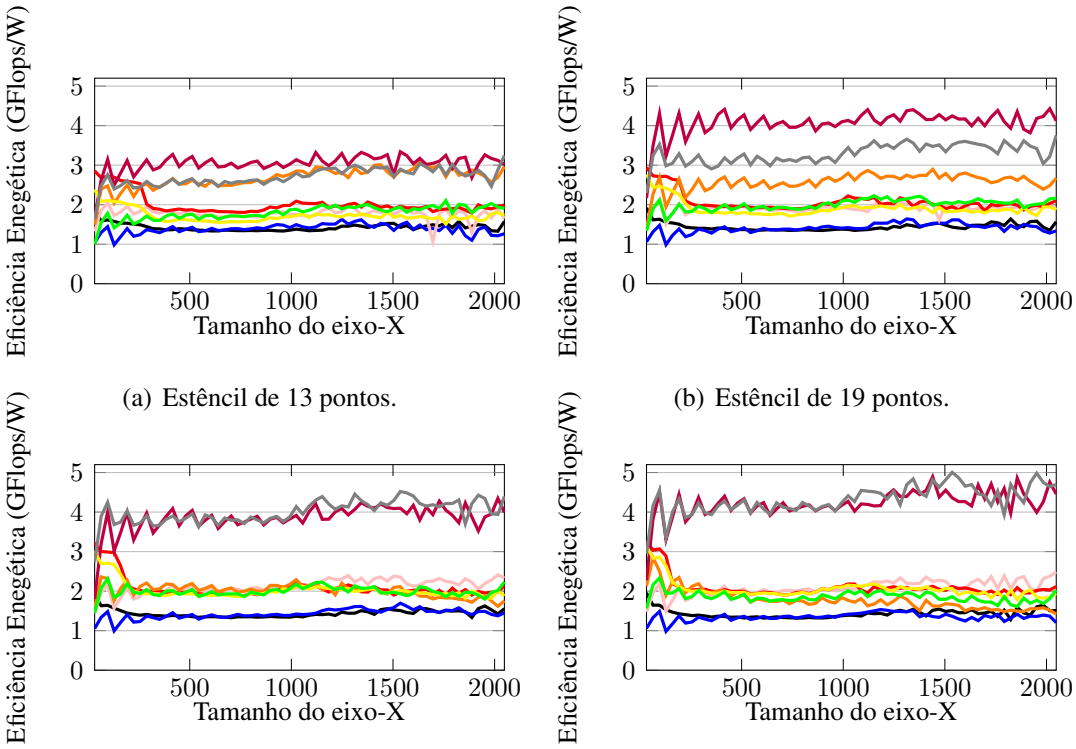
(a) Estêncil de 13 pontos.

(b) Estêncil de 19 pontos.

(c) Estêncil de 25 pontos.

(d) Estêncil de 31 pontos.

Figura 1. Desempenho de diferentes tamanhos de stencil.



(a) Estêncil de 13 pontos.

(b) Estêncil de 19 pontos.

(c) Estêncil de 25 pontos.

(d) Estêncil de 31 pontos.

Figura 2. Eficiência energética obtida por cada aplicação.

5. Conclusão

A computação de estênceis apresenta baixa intensidade computacional, uma vez que estas aplicações são tipicamente *memory-bound*. Desta forma, otimizações de memória são importantes para utilizar as memórias mais rápidas disponíveis na GPU e aumentar sua eficiência energética.

Neste artigo, aplicamos otimizações para aplicações de estêncil que usam diferentes níveis de memória de GPUs, com a finalidade de analisar o impacto do uso dos diferentes níveis de memória no desempenho e na eficiência energética das aplicações. As otimizações aplicadas permitem o uso da cache somente leitura e também o uso da memória compartilhada. Além disto, a outra otimização proposta permite a combinação da internalização do eixo-Z da aplicação de estêncil com o reuso dos registradores da arquitetura GPU.

A principal contribuição deste artigo é o aumento do desempenho e o aumento da eficiência energética de aplicações de estêncil por meio da melhoria do algoritmo e da melhoria do acesso a memória. Nossos algoritmos de computação de estênceis otimizados para GPUs obtiveram um aumento de desempenho de até $2,85$. Os resultados computacionais também destacam uma economia de energia de cerca de 20%. Além disso, usando diferentes cargas de trabalho de estêncil, nossos métodos e otimização aumentam a eficiência energética em até 50%.

Mudanças na arquitetura de GPU, como no caso da introdução da cache somente leitura na arquitetura Kepler, podem gerar mudanças nos resultados apresentados neste trabalho. Em trabalhos futuros, planejamos investigar métodos e otimizações para para obter ganhos em aplicações de estêncil sobre novas arquiteturas GPU e também para arquiteturas Intel Xeon Phi.

Referências

[Bergman et al. 2008] Bergman, K., Borkar, S., Campbell, D., Carlson, W., Dally, W., Denneau, M., Franzon, P., Harrod, W., Hill, K., Hiller, J., et al. (2008). Exascale computing study: Technology challenges in achieving exascale systems. *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep*, 15:1–297.

[de la Cruz and Araya-Polo 2011] de la Cruz, R. and Araya-Polo, M. (2011). Towards a multi-level cache performance model for 3d stencil computation. *Procedia Computer Science*, 4:2146–2155.

[Hamilton et al. 2015] Hamilton, B., Webb, C. J., Gray, A., and Bilbao, S. (2015). Large stencil operations for gpu-based 3-d acoustics simulations. *Proc. Digital Audio Effects (DAFx),(Trondheim, Norway)*, pages 292–299.

[Maruyama and Aoki 2014] Maruyama, N. and Aoki, T. (2014). Optimizing stencil computations for nvidia kepler gpus. In *Proceedings of the 1st International Workshop on High-Performance Stencil Computations, Vienna*, pages 89–95.

[Nasciutti and Panetta 2016] Nasciutti, T. C. and Panetta, J. (2016). Impacto da arquitetura de memória de gpgpus na velocidade da computação de estênceis. *XVII Simpósio de Sistemas Computacionais*, pages 64–75.