

Avaliação de Desempenho de uma Aplicação de Estêncil 3D em Plataformas NVIDIA GH200/CUDA e AMD MI300A/HIP

Laura B. Ludgero¹, Helio Crestana Guardia¹, Hermes Senger²

¹Departamento de Computação – Universidade Federal de São Carlos (UFSCar)

²Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (ICMC/USP)
São Carlos – SP – Brasil

lauraludgero@estudante.ufscar.br, helio.guardia@ufscar.br, hsenger@icmc.usp.br

Abstract. *This work evaluates the performance of MiniWave, a 3D stencil application, on AMD Instinct MI300A (HIP) and NVIDIA GH200 (CUDA). Experiments investigate the impact of problem size and thread-block configuration on execution time and throughput. Results indicate that performance depends strongly on block organization and differs across architectures due to their execution models.*

Resumo. *Este trabalho avalia o desempenho da aplicação MiniWave, baseada em estêncil 3D, em GPUs AMD Instinct MI300A (HIP) e NVIDIA GH200 (CUDA). Os experimentos investigam o impacto do tamanho do problema e da configuração dos blocos no tempo de execução e no throughput. Os resultados indicam que o desempenho depende fortemente da organização dos blocos e varia entre arquiteturas devido aos seus modelos de execução.*

1. Introdução

GPUs são amplamente utilizadas em computação de alto desempenho (HPC) devido ao elevado paralelismo e à alta largura de banda de memória [Navaux et al. 2023]. Entretanto, o desempenho depende das características da arquitetura e do ambiente de execução.

Aplicações baseadas em *stencil* são comuns em simulações científicas por apresentarem elevado paralelismo de dados e forte dependência da movimentação de memória [Datta et al. 2009]. Este trabalho avalia o desempenho da aplicação MiniWave, baseada em *stencil* 3D, em duas plataformas: AMD Instinct MI300A com HIP [Advanced Micro Devices 2024] e NVIDIA GH200 com CUDA [NVIDIA 2024], considerando diferentes tamanhos de problema e configurações de blocos.

2. Metodologia

A aplicação utilizada foi o MiniWave, um simulador de propagação de ondas acústicas tridimensionais baseado no framework SimWave [HPCSys-Lab 2025]. O modelo emprega Diferenças Finitas no Tempo e no Espaço (FDTD), resultando em um padrão computacional do tipo *stencil* 3D de 7 pontos. A atualização do campo de onda é dada por:

$$u_{i,j,k}^{t+1} = 2u_{i,j,k}^t - u_{i,j,k}^{t-1} + v^2 dt^2 \nabla^2 u_{i,j,k}^t \quad (1)$$

em que $u_{i,j,k}^t$ representa o campo de onda na posição (i, j, k) no instante t , v a velocidade de propagação, dt o passo de tempo e $\nabla^2 u_{i,j,k}^t$ o Laplaciano espacial discretizado pelo *stencil* de 7 pontos.

Os experimentos foram conduzidos no supercomputador Santos Dumont (SDumont), utilizando implementações equivalentes em HIP (ROCm 7.0.2) e CUDA (12.6). A implementação HIP foi compilada com `hipcc`, enquanto a implementação CUDA utilizou o `nvcc` 12.6, com arquitetura alvo `sm_90` e as opções `-O3` e `--use_fast_math`. Em ambas as plataformas, o CMake foi configurado em modo `Release`. Foram avaliadas malhas de 128^3 , 256^3 , 512^3 e 1024^3 , considerando configurações de bloco 1^3 , 8^3 , $16 \times 8 \times 8$ e $16 \times 16 \times 4$. Cada configuração foi executada três vezes, com 200 passos de tempo por execução, sendo apresentados os valores médios. A corretude foi verificada utilizando a execução em CPU como referência, com erro absoluto máximo de $2,21 \times 10^{-5}$ para CUDA e $2,81 \times 10^{-5}$ para HIP, sem valores superiores a 10^{-4} .

O desempenho foi avaliado pelo tempo de execução e pelo *throughput* (Mpts/s), calculado como:

$$\text{Throughput} = \frac{N_x N_y N_z N_{\text{steps}}}{T \cdot 10^6}, \quad (2)$$

em que N_x , N_y e N_z são as dimensões da malha, N_{steps} o número de passos de tempo e T o tempo de execução, em segundos. O tempo inclui alocação e liberação de memória na GPU, transferências host-device, execução e sincronização dos kernels, excluindo operações de E/S em HDF5. O fator 10^6 expressa o resultado em Mpts/s. Adicionalmente, realizou-se o *profiling* da configuração 128^3 com blocos $8 \times 8 \times 8$, utilizando NVIDIA Nsight Systems e Nsight Compute na GH200/CUDA e `rocprofv3` na MI300A/HIP, para analisar a ocupação, o tempo de execução do kernel e as operações de cópia entre *host* e dispositivo.

3. Resultados e Análise

3.1. Implementações e Ambiente Experimental

As implementações CUDA e HIP mantêm a mesma lógica algorítmica e formulação numérica, diferindo nas interfaces específicas de cada ecossistema. Os Algoritmos 1 e 2 apresentam seus fluxos de execução, incluindo alocação e transferência de memória, execução do kernel e sincronização.

Algorithm 1 Implementação CUDA	Algorithm 2 Implementação HIP
1: Alocar memória com <code>cudaMalloc</code>	1: Alocar memória com <code>hipMalloc</code>
2: Copiar dados com <code>cudaMemcpy</code>	2: Copiar dados com <code>hipMemcpy</code>
3: for cada Passo de Tempo do	3: for cada Passo de Tempo do
4: <code>wave_kernel<<<...>>></code>	4: <code>hipLaunchKernel...</code>
5: end for	5: end for
6: <code>cudaDeviceSync...</code>	6: <code>hipDeviceSync...</code>
7: Kernel:	7: Kernel:
8: Calcular índices (i, j, k)	8: Calcular índices (i, j, k)
9: if fora da região válida then	9: if fora da região válida then
10: Encerrar thread	10: Encerrar thread
11: end if	11: end if
12: Ler vizinhos e stencil	12: Ler vizinhos e stencil
13: Atualizar resultado	13: Atualizar resultado

Embora a estrutura de alto nível seja equivalente, o comportamento em execução depende diretamente da arquitetura utilizada. A Tabela 1 resume as principais características das plataformas empregadas nos experimentos.

Tabela 1. Características das plataformas avaliadas

Característica	NVIDIA GH200 120GB	AMD Instinct MI300A
Arquitetura GPU	Hopper	CDNA3 (gfx942)
Unidades de processamento	132 SMs	228 CUs
Granularidade de execução	Warp (32 threads)	Wavefront (64 threads)

As plataformas possuem organizações internas distintas: a GH200 utiliza SMs e warps de 32 threads, enquanto a MI300A emprega CUs e wavefronts de 64 threads. Assim, um bloco $8 \times 8 \times 8$, com 512 threads, corresponde a 16 warps na GH200 e 8 wavefronts na MI300A. Essa diferença na granularidade de execução pode contribuir para o comportamento distinto das configurações de bloco nas duas arquiteturas.

A variabilidade entre as três execuções foi baixa: o coeficiente de variação máximo observado foi de aproximadamente 2,28% na MI300A/HIP e 3,88% na GH200/CUDA, indicando estabilidade dos resultados apresentados.

As Tabelas 2 e 3 evidenciam forte dependência do desempenho em relação à configuração de bloco. Em ambas as plataformas, $1 \times 1 \times 1$ apresentou o menor desempenho, enquanto $8 \times 8 \times 8$ e $16 \times 8 \times 8$ produziram os maiores *throughputs*.

Tabela 2. Tempo de execução e throughput da implementação HIP na MI300A

Size	1x1x1		8x8x8		16x8x8		16x16x4	
	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s
128	0.473963	885.25	0.278398	1506.64	0.278569	1505.66	0.283608	1479.10
256	1.867102	1797.14	0.361399	9284.76	0.365468	9181.46	0.397560	8440.62
512	13.139648	2042.94	0.993441	27020.81	0.984518	27265.83	1.220666	21991.06
1024	103.870454	2067.46	6.130048	35032.18	6.187368	34708.07	8.039768	26710.80

Tabela 3. Tempo de execução e throughput da implementação CUDA na GH200

Size	1x1x1		8x8x8		16x8x8		16x16x4	
	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s	Tempo s	Throughput Mpts/s
128	0.574488	730.34	0.345309	1215.03	0.344373	1218.50	0.356774	1176.81
256	2.318394	1447.33	0.371431	9042.38	0.391743	8568.07	0.413948	8106.07
512	16.289770	1647.88	0.655310	40977.17	0.820891	32700.92	0.944680	28416.87
1024	128.687159	1668.76	2.979036	72087.39	4.162342	51593.34	5.114666	41989.16

3.2. Comparação de Desempenho entre as Plataformas

A Figura 1 compara os *throughputs* das plataformas. Para 128^3 e 256^3 , os resultados são da mesma ordem de grandeza; com o aumento da malha, a GH200/CUDA passa a apresentar maior desempenho, atingindo aproximadamente 41 contra 27 Gpts/s em 512^3 e 72 contra 35 Gpts/s em 1024^3 .

Para investigar esse comportamento, realizou-se o *profiling* da configuração 128^3 com blocos $8 \times 8 \times 8$. A Tabela 4 apresenta a ocupação média do kernel, seu tempo acumulado nas 200 execuções e os tempos das operações de cópia entre *host* e dispositivo registradas pelos *profilers*.

O *profiling* indicou ocupação de 56,15% na GH200 e 49,98% na MI300A. O tempo acumulado do kernel foi 3,783 ms na GH200 e 9,133 ms na MI300A, uma diferença de aproximadamente 2,4×, enquanto as operações de cópia registradas totalizaram 0,177 ms e 0,744 ms, respectivamente. Como as métricas de ocupação são obtidas por ferramentas específicas de cada ecossistema, sua comparação deve ser interpretada com cautela. Ainda assim, os tempos evidenciam diferenças no comportamento de execução das plataformas, sem permitir atribuí-las isoladamente ao modelo de programação.

Tabela 4. Resultados de *profiling* para 128^3 e blocos $8 \times 8 \times 8$.

Plataforma	Ocup. (%)	Kernel total (ms)	H→D (ms)	D→H (ms)
GH200/CUDA	56,15	3,783	0,074	0,103
MI300A/HIP	49,98	9,133	0,448	0,296

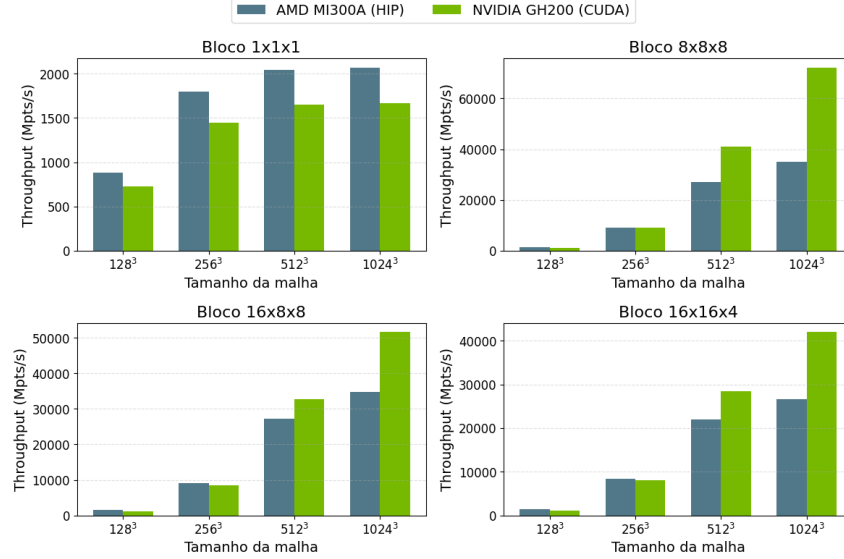


Figura 1. Throughput obtido nas plataformas GH200/CUDA e MI300A/HIP.

4. Conclusão

Este trabalho comparou o desempenho do MiniWave nas plataformas AMD MI300A/HIP e NVIDIA GH200/CUDA. Os resultados mostraram forte influência da configuração de blocos, com a GH200/CUDA alcançando aproximadamente $2 \times$ o *throughput* da MI300A/HIP na malha 1024^3 . O *profiling* indicou, para a configuração 128^3 com blocos $8 \times 8 \times 8$, menor tempo de execução do kernel e das operações de cópia na GH200/CUDA, além de diferenças na ocupação das GPUs. Como hardware e ecossistema de software variaram simultaneamente, as diferenças observadas não podem ser atribuídas isoladamente a CUDA ou HIP. Como trabalho futuro, pretende-se investigar otimizações visando maior portabilidade de desempenho entre arquiteturas de GPU.

Referências

- Advanced Micro Devices (2024). Hip documentation (rocm platform). Acessado em: 25 mar. 2026.
- Datta, K., Kamil, S., Williams, S., Oliner, L., Shalf, J., and Yelick, K. (2009). Optimization and performance modeling of stencil computations on modern microprocessors. *SIAM Review*, 51(1):129–159.
- HPCSys-Lab (2025). Simwave: Acoustic wave simulation framework. Acessado em: 26 mar. 2026.
- Navaux, P. O. A., Lorenzon, A. F., and da Silva Serpa, M. (2023). Challenges in high-performance computing. *Journal of the Brazilian Computer Society*, 29(1).
- NVIDIA (2024). Cuda toolkit documentation. Acessado em: 25 mar. 2026.