

# Avaliação de Placa Gráfica Utilizando simulação de N-CORPOS em CUDA

Gustavo Ebbo<sup>1</sup>, Ivo S. Paiva<sup>2</sup>, Rodrigo V. Calábria<sup>3</sup>, Marcelo Zamith<sup>4</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade Federal Rural do Rio de Janeiro (UFRRJ) – Nova Iguaçu – RJ – Brasil

{gustavoebbo, ivopaiva, rodrigovcalabria, mzamith}@ufrrj.br

**Abstract.** *Several applications try to test GPUs' features, such as memory and processors. However, these applications whether present no-understandable metrics or they don't work as expected, making users frustrated or confused. Therefore, this work presents an application to test GPUs' features and provides simple valuation metrics.*

**Resumo.** *Varias aplicações buscam testar as características da GPUs, tais como memórias e processadores. Entretanto, estas aplicações ou apresentam métricas incompreensíveis ou não funcionam corretamente, frustrando os usuários. Portanto, este trabalho apresenta uma aplicação para testar as características da GPU e fornece métricas simples de avaliação.*

## 1. Introdução

Avaliações de hardwares exigem que as aplicações utilizadas evidenciem, de forma objetiva e clara, os hardwares que estão sendo testados. Além disso, como os hardwares gráficos (GPUs) são dispositivos com capacidade de processamento massivamente paralelo, ou seja, exigem uma massa grande de dados, aplicações utilizadas como testes devem proporcionar uma massa de dados e operações computacionais que demonstrem a real capacidade da GPU com clareza.

Neste sentido, é proposto no trabalho uma aplicação de simulação de n-corpos para avaliar a performance das GPUs baseadas em CUDA. Além disso, é apresentado uma função de pontuação para definir a capacidade da GPU testada. O objetivo do uso da função é mostrar de forma simples e clara, para quem testa, o quão boa é a GPU testada e a comparação desta com demais GPUs, inclusive de outras arquiteturas, que dispõe de processadores e barramento mais rápidos além de uma quantidade maior de *cores*.

A simulação de n-corpos busca reproduzir os fenômenos da interação entre galáxias [Playne et al. 2009], onde cada partícula está sob a influência das leis gravitacionais de Newton. As simulações apresentadas neste trabalho empregam o método *All-Pairs* [Couchman 1991], que consiste em um algoritmo que calcula a força que cada partícula exerce sobre as demais partículas. É considerado o mais simples por utilizar uma técnica de força bruta, mas apresentar complexidade quadrática. É um método estável quando comparado com outros métodos de menor complexidade ( $O(N \log N)$ ). Por fim, adotou-se as métricas apresentadas por Playne et al. [Playne et al. 2009].

**Métricas do benchmark:** A avaliação da aplicação considerando a GPU utiliza a ideia de pontuação. Para isto, são adotadas duas funções: *i*) a Eq. 1 que visa avaliar a

performance do hardware em relação às instâncias, considerando a complexidade do Algoritmo, onde  $\Delta T$  é o intervalo de tempo gasto para processar os  $n$ -corpos em segundos; e, *ii*) a Eq. 2 considera características do hardware a fim de comparar o desempenho da aplicação em diferentes GPUs, onde  $P$  é a quantidade de processadores e  $C$  é o *clock* destes processadores.

$$f(N) = \sum_{N=1}^K \frac{N^2}{\Delta T} \quad (1)$$

$$g(N, P, C) = \frac{1}{P} \times \frac{500}{C} \times f(N) \quad (2)$$

A avaliação do hardware segundo estas métricas requer que quanto maior a pontuação obtida com as Eqs. 1 e 2 melhor é o resultado, incluindo um melhor aproveitamento dos recursos da GPU.

**Implementação em CUDA:** A implementação foi desenvolvida utilizando a biblioteca CUDA da Nvidia. A granularidade do processamento é de uma partícula por *thread* e esta é responsável pelo cálculo das forças aplicadas na partícula.

Como o método *All-Pairs* requer a influência de todas as partículas do sistema para o cálculo da nova posição de cada partícula, foi adotada na implementação o uso da memória compartilhada.

**Testes e resultados:** Para os testes utilizados neste trabalho foram empregados a seguinte configuração: para  $N = [1024, 2048, 4096, 8192]$ , onde  $N$  são os corpos no sistema, foram simulados 1.000 passos de tempo. Para  $N = [16.384, 32.768]$  foram executados 250 passos. A tabela 1 mostra as GPUs empregadas, suas características e os resultados alcançados. Observa-se que a GPU com mais poder computacional obteve um *score* 2,7 maior em relação a GPU mais fraca (Nvidia GT 740M). Por outro lado, a NVidia GT 740M obteve uma pontuação superior na *efficiency*. Estudos preliminares indicam que este resultado ocorreu devido a um menor custo de gerencia da placa com menos cores. **Conclusão e trabalhos futuros:** Os resultados obtidos com as Eq. 1 e 2 são

**Tabela 1. Resultados**

	CUDA cores	Freq.	Score	Efficiency
NVidia GT 740M	384	900	7.452.884, 50	9.902, 32
NVidia GTX 970	1.664	1, 178	20.496.760, 00	5.528, 62

de simples interpretação e evidenciam a capacidade de operações de ponto flutuante da GPU. Contudo, os testes apresentaram um resultado onde a placa com mais capacidade de processamento teórico teve uma pior utilização dos seus recursos. Este fato requer investigações mais apuradas a fim de determinar que fatores estão influenciando este resultado. Assim, como trabalhos futuros é investigar este resultado utilizando a ferramenta CUDA profiler.

## Referências

- Couchman, H. (1991). Mesh-refined p3m-a fast adaptive n-body algorithm. *The Astrophysical Journal*, 368:L23–L26.
- Playne, D. P., Johnson, M., and Hawick, K. A. (2009). Benchmarking gpu devices with n-body simulations. In *CDES*, pages 150–156.