

# Utilização de GPU na resolução do problema de absorção de água pelas raízes das plantas no contexto de irrigação de solos

Lucas Holz<sup>1</sup>, Marcos Ketzer<sup>1\*</sup>, Paulo Sausen<sup>1</sup>, Edson L. Padoin<sup>1</sup>, Philippe O. A. Navaux<sup>2</sup>

<sup>1</sup>Universidade Regional do Noroeste do Estado do Rio G. do Sul (UNIJUÍ) – Ijuí – RS – Brasil  
Grupo de Automação Industrial e Controle (GAIC)

<sup>2</sup>Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre – RS – Brasil

{lucas, ketzer, sausen, padoin}@unijui.edu.br, navaux@inf.ufrgs.br

## 1. Introdução

O aumento de desempenho durante muitos anos esteve baseado em incrementos de frequência de operação. Após esse período passou-se a utilizar arquiteturas paralelas e heterogêneas almejando maiores aumentos de eficiência e desempenho. Neste contexto, uma das alternativas que atualmente é empregada consiste na utilização de unidades de processamento gráfico (GPU) também denominados de aceleradores paralelos para computações de propósito geral (GPGPU). Alternativa esta, motivada pelos custos mais baixos, menor consumo de energia e o grande poder de processamento paralelo destes processadores gráficos.

Para atender a grande demanda de processamento gráfico, as GPUs passaram por uma grande evolução. Os modelos atuais apresentam um elevado grau de paralelismo, uma maior largura de banda de memória e uma precisão dupla superando em muitos modelos o poder computacional de muitos processadores modernos. Desta forma, tais placas aceleradoras vem ganhando destaque e tem sido empregadas na construção de novos equipamentos e em vários campos de pesquisa, como por exemplo: previsão do tempo, criptografia, algoritmos de alinhamento de sequência, simulação de dinâmica de fluidos, dentre outros[NVIDIA 2010].

Enquanto construção de equipamentos, pode-se perceber o aumento significativo do número de computadores equipados com placas GPU que são relacionadas na top500, esta que classifica os 500 maiores computadores em termos de poder computacional. Destaque ao equipamento topo da lista atual de 2.57 petaflop/s que também emprega placas GPU em sua arquitetura.

Neste contexto, este trabalho consiste na exploração do poder computacional das GPUs através da utilização da tecnologia GPU e Compute Unified Device Architecture (CUDA) na resolução do algoritmo que descreve o problema da irrigação de solo e da absorção da água pelas raízes das plantas[Padoin et al. 2006].

## 2. Plataforma CUDA

CUDA [NVIDIA 2010] é a arquitetura de computação paralela da NVIDIA que possibilita aumentos significativos na performance de computação pelo aproveitamento da potência da GPU [Humphreys and Luebke 2007]. Assim, para atender algumas demandas a tecnologia da computação está evoluindo para um processamento realizado na CPU e na GPU.

Esta plataforma se utiliza dos processadores da placa de vídeo para cálculos, podendo entregar resultados com grande precisão quando utilizado ponto flutuante. Tais processadores são nominalmente chamados de CUDA cores, sendo que, as limitações de utilização de todo potencial da plataforma CUDA esbarram na quantidade de memória e na quantidade de núcleos CUDA integrados na GPU [NVIDIA 2010]. Destaca-se que cada placa gráfica possui 3 memórias, uma interna e compartilhada entre seus núcleos SP além das duas memórias cache.

Outra característica importante que deve ser observada no desenvolvimento de aplicações para esta plataforma é o modelo de programação composto por dois princípios básicos: Streams e Kernels. O primeiro composto pelos fluxos de dados que alimentam os Kernels, estes que realizam o processamento e geram os resultados. Ou seja, enquanto uma thread para CPU pode ser exemplificada como uma tarefa que processa dados, na GPU a thread é o dado em si, onde o algoritmo que processa os dados é chamado de kernel.

### 3. Definição do problema

O problema proposto neste trabalho é a paralelização do algoritmo sequencial, cuja finalidade é estimar a quantidade de água retirada do solo pelas raízes das plantas, com base em dados do monitoramento do teor de água do solo [Borges and Klamt. 2006]. O algoritmo e o modelo matemático que descrevem o problema foram apresentados em [Padoin et al. 2006] [Padilha. et al. 2008].

O algoritmo apresenta dois fatores que justificam a grande demanda computacional:

- o número ótimo de células da malha necessário para obter resultados com a precisão desejada; e
- o elevado número de execuções do problema direto [Padilha. et al. 2008].

Com o objetivo de melhores desempenhos e uma melhor precisão dos resultados, uma nova implementação foi realizada utilizando os recursos de processamento paralelo das GPUs utilizando a plataforma CUDA.

Considerando a hipótese de que a absorção dependente do tempo e é limitada pela oferta de água do solo, foi proposta a equação (1) que considera a variação das condições climáticas cíclicas normais do período diário.

$$S = \beta(\theta - \theta_r)^{b(t-c)}$$

onde S é a taxa de absorção de água pela raiz ( $cm^3/h$ )

$\beta$  é a constante de proporcionalidade ( $cm^3/h$ )

c é o instante de tempo da máxima absorção diária (h)

t é o tempo (h)

$\theta$  e  $\theta_r$  são o teor de água no ápice e residual, respectivamente (adimensional).

### 4. Implementação Paralela e ambiente de teste

Os dados experimentais e resultados da implementação sequencial demandam elevados tempos de processamento mesmo em uma malha com poucos pontos. O problema já

foi resolvido através de uma implementação paralela em um aglomerado homogêneo composto por 20 máquinas. A eficiência, bem como os resultados foram publicados em [Padoin et al. 2006]. Neste trabalho, almejando a utilização do alto poder de processamento das GPUs através da biblioteca CUDA e com o objetivo de obtenção de resultados mais precisos com malhas maiores, devido ao crescimento exponencial do tempo de execução, uma nova implementação paralela foi desenvolvida utilizando sistema operacional GNU/Linux Ubuntu 10.04 de 64 bits, com compilador GCC versão 4.4 e plataforma CUDA/SDK versão 3.2.

O computador utilizado possui um processador Intel Core i7 720QM, 2,8GHz, 6 GB de memória RAM DDR3 e uma placa de vídeo nVidia GeForce GT 230M com 1 GB de memória.

O driver de vídeo utilizado foi a versão 260.19.14 para Linux 64 bits da NVIDIA.

Na tabela 1 é apresentado um comparativo das características da CPU e da GPU utilizadas:

**Tabela 1. Características da CPU e da GPU utilizadas**

	CPU	GPU
Processador	Intel Core i7 720QM	NVIDIA GeForce GT 230M
Modelo	Clarksfield	GT216M
Frequência de Clock (GHz)	1,6 até 2,8	1,1
Memória DDR3 (GB)	6	1
Largura de Banda Máxima (GB/s)	21	25
Velocidade de Memória (MHz)	800	800
Número de Núcleos	4	48
Largura do Bus (bits)	64	128
Consumo Potencia Máx (Watts)	45	25

## 5. Resultados

Esta seção apresenta os resultados obtidos na execução do algoritmo paralelo. A análise de desempenho da implementação paralela foi realizada através da execução paralela com diferentes ordens de matrizes.

Na Tabela 2 é apresentado os tempos médios de dez execuções em segundos e o desvio padrão obtidos com a execução do algoritmo paralelo e sequencial com matrizes de ordens diferentes.

**Tabela 2. Tempo médio de execução do algoritmo na CPU e na GPU (em seg)**

Ordem Matriz	CPU	GPU
8 x 8	0.818 ± 0.063	0.493 ± 0.012
16 x 16	4.349 ± 0.060	1.332 ± 0.016
32 x 32	4.139 ± 0.081	1.301 ± 0.022
64 x 64	9.458 ± 0.119	2.015 ± 0.010
128 x 128	29.437 ± 0.215	7.770 ± 0.022
256 x 256	168.435 ± 0.125	40.809 ± 0.042

Observado a Tabela 2, percebe-se que os tempos de execução do algoritmo na GPU foi sempre menor que o obtido nas execuções na CPU. O desempenho da GPU com matrizes a partir de 16, para esta aplicação foi superior a 3 vezes, assim motivando novas implementações do algoritmo nesta arquitetura.

## 6. Conclusão

Este artigo apresentou a análise de desempenho da arquitetura GPU com a plataforma CUDA através da paralelização do algoritmo que descreve o problema da irrigação de solo e da absorção da água pelas raízes das plantas.

A paralelização do problema é justificada devido a grande demanda computacional necessária para resolução do mesmo além da necessidade de obtenção de resultados mais precisos. Tal algoritmo já havia sido paralelizado e executado em um aglomerado de computadores, porém, nesta nova versão paralela desenvolvida para arquitetura GPGPU obteve-se tempos de execução 3 vezes menores que a versão sequencial.

Assim, o desenvolvimento de programas paralelos é motivado não apenas pelo ganho de desempenho que se obtém, mas também pela possibilidade de exploração de novas alternativas para resolução de problema, principalmente pelo aumento da precisão dos resultados.

Com a utilização de processamento paralelo conseguiu-se resolver o sistema em menores tempos de execução utilizando malhas compostas de um maior número de pontos o que conseqüentemente aumentou a precisão dos resultados. Os resultados obtidos permitem que trabalhos futuros possam ser desenvolvidos no intuito de validar novos testes na área de irrigação de solos sem a necessidade da execução de rotinas que demandem de grandes tempos de execução.

## 7. Agradecimentos

Os autores agradecem o apoio financeiro (bolsa de Iniciação Científica\*) concedido pela Fundação de Amparo a Pesquisa do Estado do Rio Grande do Sul (FAPERGS).

## Referências

- Borges, P. A. P. and Klamt., V. (2006). Modelagem matemática da retirada de água do solo pelas raízes de plantas. In *XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional*.
- Humphreys, D. and Luebke, G. (2007). How gpus work. In *IEEE Computer*, volume 40, pages 96–100.
- NVIDIA (2010). *NVIDIA CUDA C Programming Guide 3.2*. Disponível em <http://developer.download.nvidia.com/compute/cuda>.
- Padilha., N. R., Dill, S. L., Borges, P. A., and Padoin, E. L. (2008). Resolução do problema de absorção da água do solo através da paralelização do problema inverso. In *VIII Escola Regional de Alto Desempenho*, pages 268–271. Evangraf.
- Padoin, E. L., Dill, S. L., Borges, P. A., and Schaedler., R. (2006). Paralelização de métodos computacionais aplicados à análise das variações do teor de umidade de solos saturados e não saturados. In *WSCAD 2006 - VII Workshop em Sistemas Computacionais de Alto Desempenho*, pages 97–104.