

Aplicação de Processamento Paralelo com GPU a Problemas de Escoamento Monofásico em Meios Porosos

Bruno Pereira dos Santos
Dany Sanchez Dominguez

Roteiro

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução
5. Resultados
6. Conclusão
7. Trabalhos Futuros

1. Introdução ←
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução
5. Resultados
6. Conclusão
7. Trabalhos Futuros

INTRODUÇÃO

Introdução

- Problemas de Computação de Alto Desempenho (CAD)
 - Modelagem e Simulação de fenômenos da natureza
 - Engenharias
 - Bioinformática
 - Física médica
- Técnicas CAD
 - Tradicionais
 - Memória compartilhada
 - Memória distribuída
 - General-Purpose Computation on Graphics Hardware (GPGPU)
 - Uso da GPU

Fontes:
[Aiping D, 2011] ,
[Alonso P. 2009],
[Goddeke D. 2007].

Fonte:
[GPGPU. 2012]

Introdução

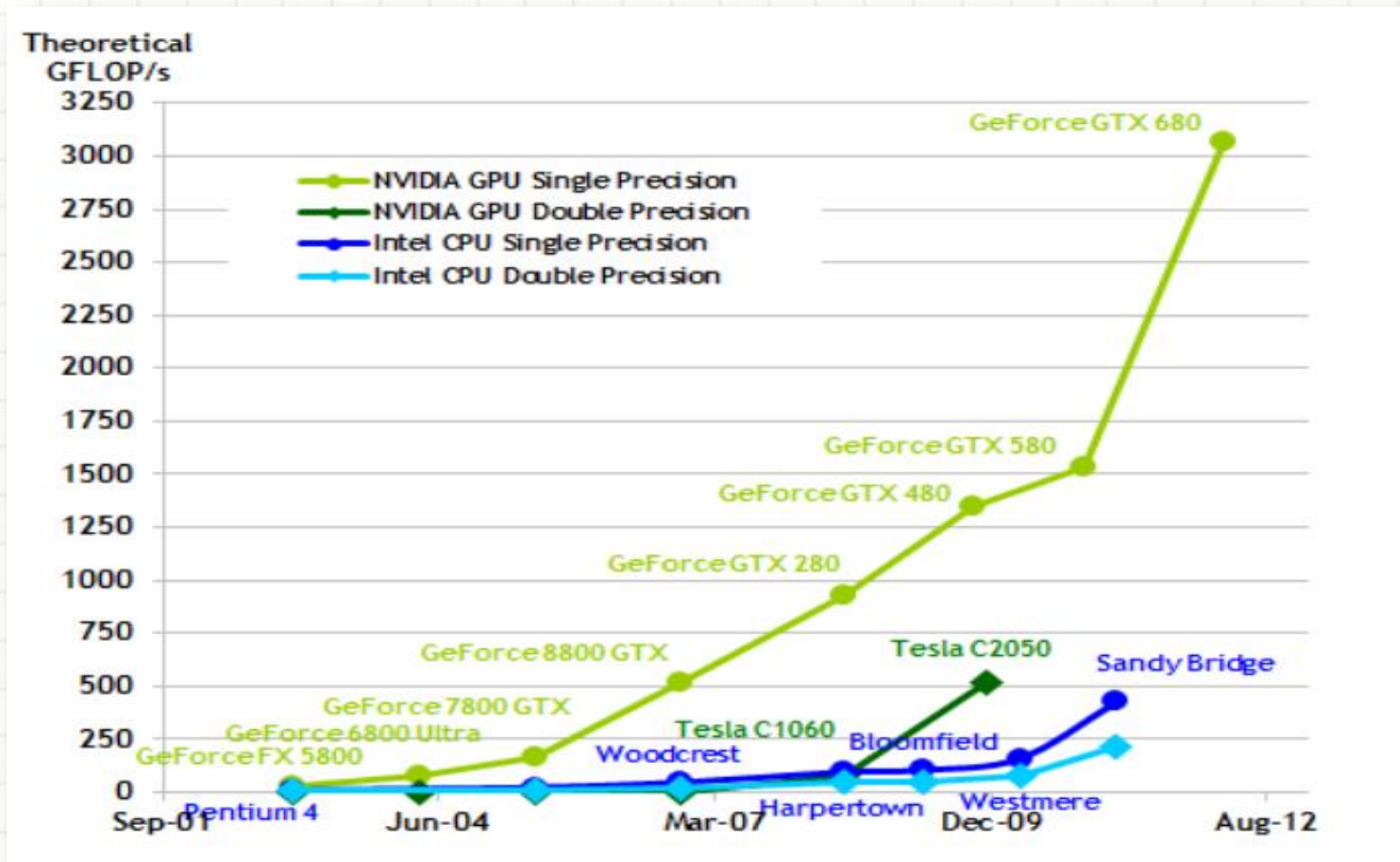


Gráfico 1 - Operações de ponto flutuante por segundo entre CPU e GPU

1. Introdução
2. **Five-Spot Problem** ←
3. Modelagem Matemática
4. Resolução
5. Resultados
6. Conclusão
7. Trabalhos Futuros

FIVE-SPOT PROBLEM

Five-Spot Problem

- O que é Five-Spot Problem?
 - É o problema que aparece ao tentarmos modelar o regime de exploração de reservatórios de petróleo

Five-Spot Problem

- Configuração do problema
 - 5 poços onde:
 - 4 periféricos
 - 1 central

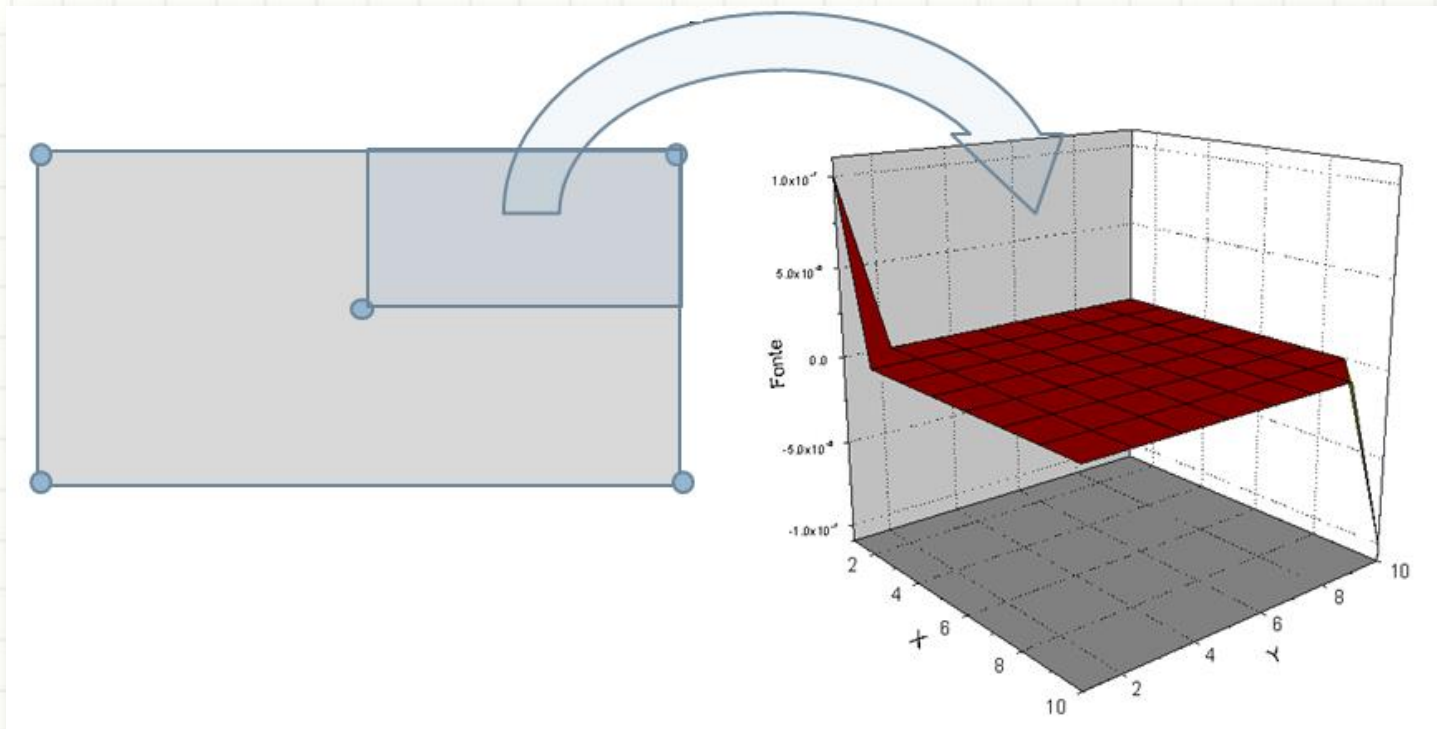


Figura 1 – Representação do problema dos cinco poços.

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática ←
4. Resolução
5. Resultados
6. Conclusão
7. Trabalhos Futuros

MODELAGEM MATEMÁTICA

Modelagem Matemática

- Eq. Diferenciais Parciais (EDP)
 - Podem descrever fenômenos com dependência espacial e/ou temporal
- Eq. de Poisson
 - É uma EDP do tipo elíptico
 - Modela bem o problema do escoamento de fluídos
 - Utilizamos geometria cartesiana bidimensional para representar o problema

Modelagem Matemática

- Eq. de Poisson

$$\vec{q}(x, y) = -k(x, y)\nabla p(x, y) \quad (1.1)$$

$$\nabla \vec{q}(x, y) = f(x, y) \quad (1.2)$$

$$\vec{q}(x, y) \cdot \mathbf{n} |_{r} = 0 \quad (1.3)$$

- Onde:

- $\vec{q}(x, y)$ é a velocidade do escoamento do fluido
- $k(x, y)$ é o coeficiente de permeabilidade do meio
- $p(x, y)$ é a pressão
- $f(x, y)$ é a função fonte

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução ←
5. Resultados
6. Conclusão
7. Trabalhos Futuros

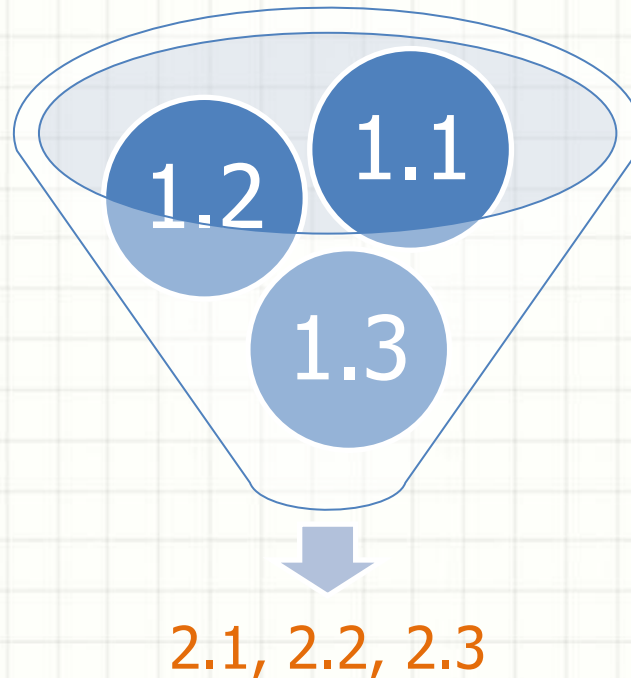
RESOLUÇÃO

Resolução

- Para resolver numericamente o sistema de equações
 - Utilizou-se o método de elementos finitos de Riviart-Thomas } Fonte: [RAVIART e THOMAS, 2007]
 - Divide-se o domínio em células quadradas de tamanho h

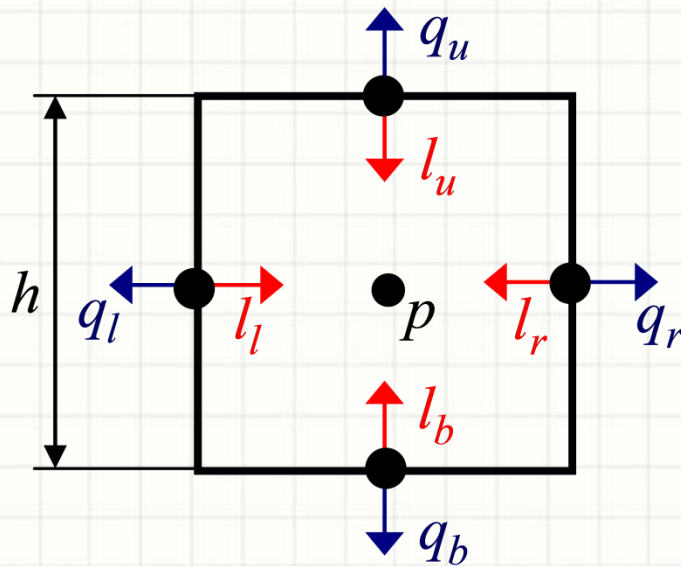
Resolução

- As equações iniciais podem ser reescritas



Resolução

- Equações reescritas



$$q_r + q_u + q_l + q_b = f \cdot h \quad (2.1)$$

$$q_\alpha = -\frac{2k}{h}(l_\alpha - p) \quad (2.2)$$

$$l_\alpha = \beta_{\alpha, \bar{\alpha}}(q_\alpha + q_{\bar{\alpha}}) + l_{\bar{\alpha}} \quad (2.3)$$

- Onde:

- α pode ser ($l = \text{left}, r = \text{right}, u = \text{up}, b = \text{bottom}$)

Resolução

- Limitação
 - Necessidade de um grande número de células
 - Consequências
 - Alto custo computacional
 - Memória
 - Processamento
 - Necessidade de CAD

Resolução

- Implementação
 - Utilizou-se
 - 1 versão serial em linguagem C
 - 1 versal paralela em linguagem C e padrão CUDA

Resolução

I. Discretização do domínio em memória

I. Carregar Mat. Fontes

II. Carregar Mat. Permeabilidade

III. Erro máx. Permitido

IV. Tamanho da região

II. Calcular betas

III. Resolução numérica da Eq de Poisson (método iterativo)

I. Chute inicial para q, p e l

II. Computar novos q e p usando os valores antigos de \bar{q} e \bar{l}

III. Verificar Convergência

Resolução

- Resolução em CUDA
 - Kernel com as dimensões do campo de exploração
 - Identificação dos threads através da API CUDA
- Fonte: [NVIDIA Corporation, 2012]

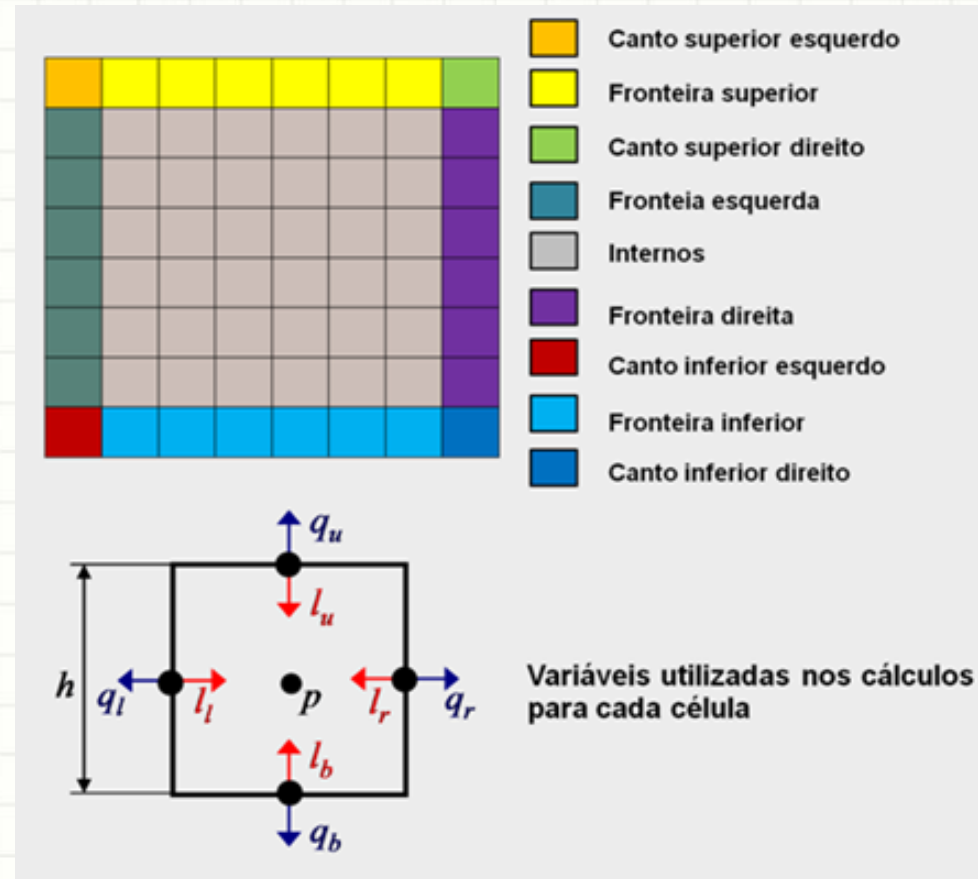


Figura 2 – Organização da matriz (domínio de cálculo)

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução
5. Resultados ←
6. Conclusão
7. Trabalhos Futuros

RESULTADOS

Resultados

- Máquina utilizada para os experimentos

Configuração da estação de trabalho

| | |
|----------------------------|---|
| Processador | Intel ® Core i7 CPU 860 2,8GHz |
| Memória RAM | 8GB |
| Placa Aceleradora Gráfica | GPU Nvidia GeForce 9800GT, 112 cores, 512 de RAM, 256bits PCI Express 16x |
| Experimentos com as verões | Serial e CUDA |

Resultados

Tempos de processamento SERIAL X PARALELO

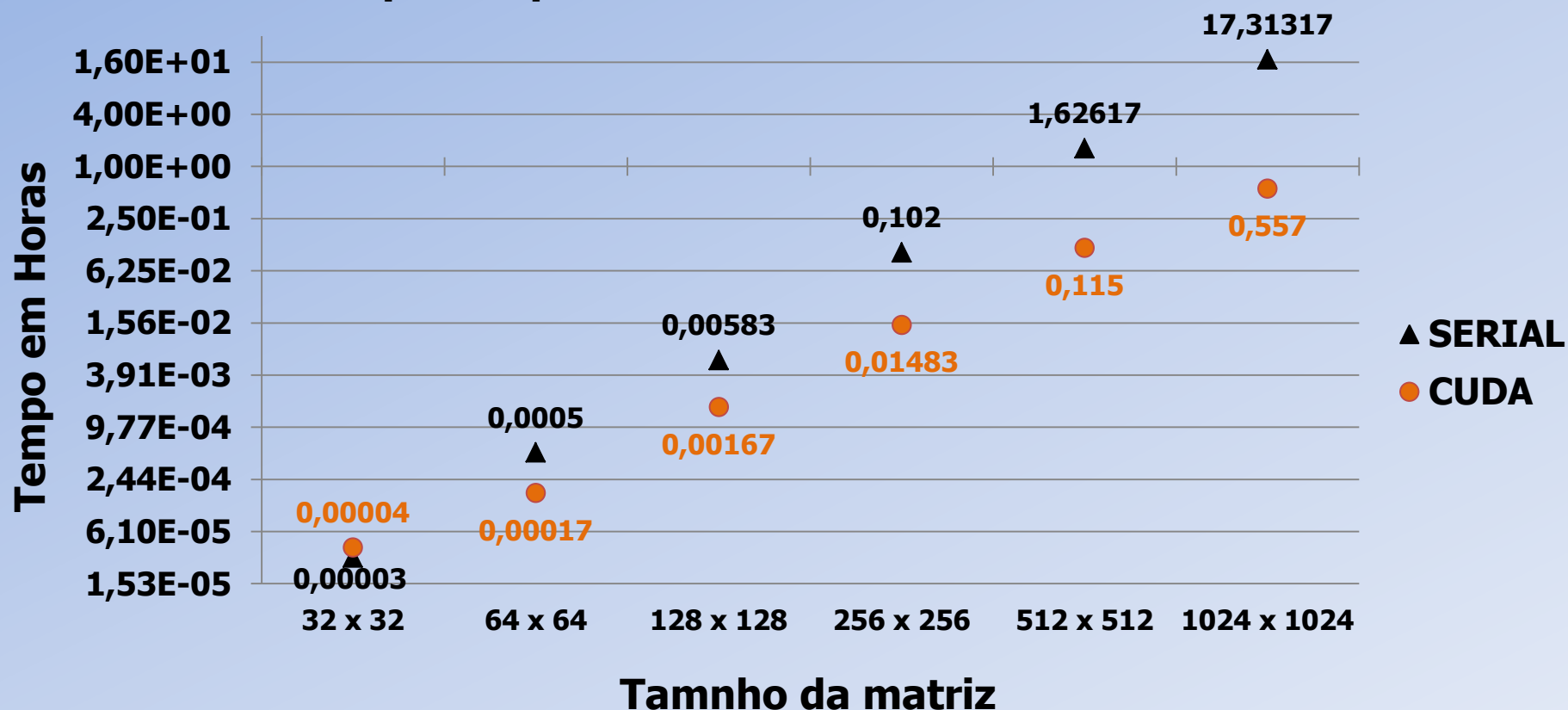


Gráfico 2 – Tempo de processamento das versões SERIAL e CUDA

Resultados

Gráfico do SPEEDUP entre Serial e Paralelo

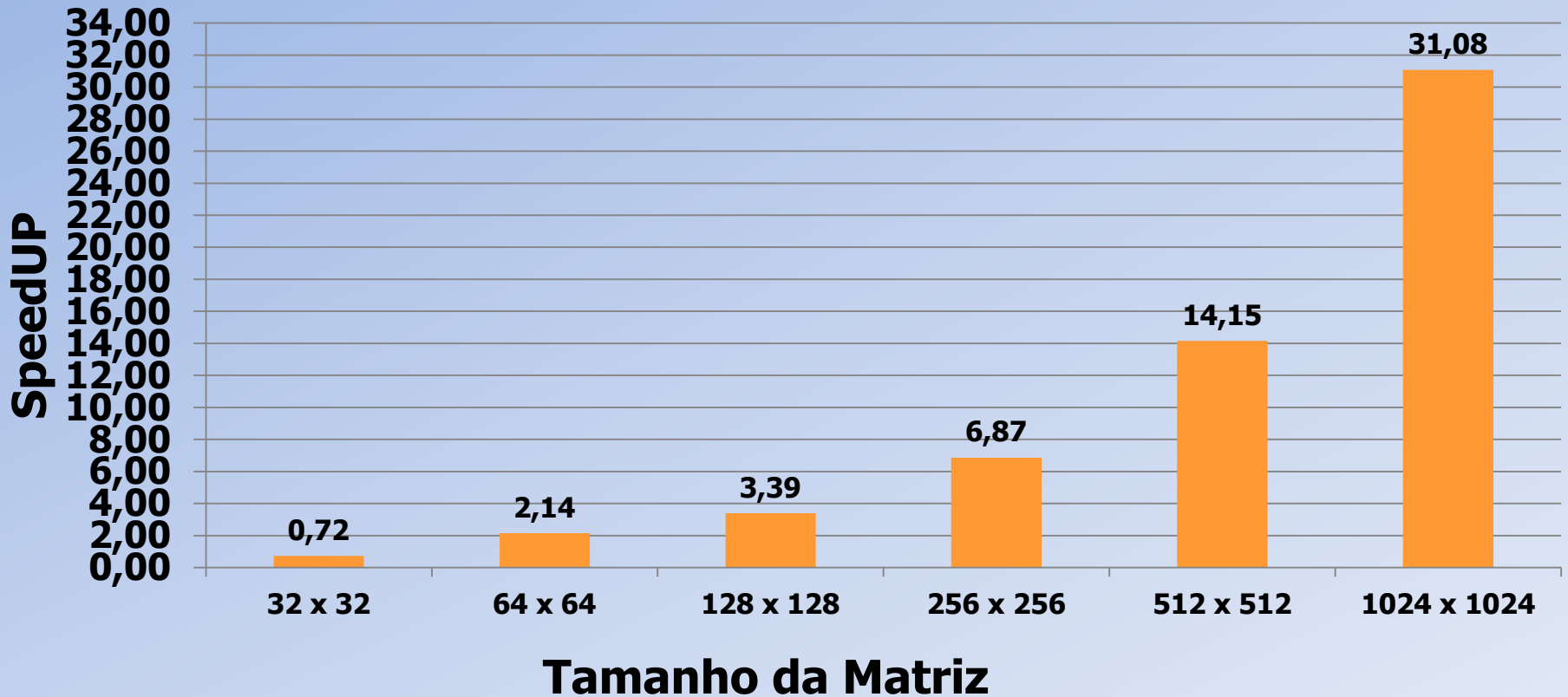


Gráfico 3 – SpeedUP entre CPU e GPU

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução
5. Resultados
6. Conclusão ←
7. Trabalhos Futuros

CONCLUSÃO

Conclusão

- Desempenho
 - CUDA x SERIAL
- Conquistas
 - Framework de padronização OpenCL
 - Difusão no meio acadêmico
 - Trabalhos Realizados
- Desafios
 - Incompatibilidade de hardware
 - Memória
 - GPU com menos recursos
 - Programação com threads
 - Novos recursos nas placas gráficas
 - CUDA – Fermi, Kepler.

1. Introdução
2. Five-Spot Problem
3. Modelagem Matemática
4. Resolução
5. Resultados
6. Conclusão
7. Trabalhos Futuros ←

TRABALHOS FUTUROS

Trabalhos Futuros

- Realizar novos comparativos com as versões tradicionais de CAD e o paradigma CUDA
- Otimizar o código CUDA existente para as novas plataformas nVidia Kepler
 - Novas possibilidades
 - Kernel dentro de Kernel
 - Chamadas recursivas
- Construir versões híbridas
 - OpenMP-CUDA, OpenMP-MPI-CUDA, CUDA-MPI

Dúvidas



Agradecimentos



Fundação de Amparo
à Pesquisa do Estado da Bahia



SECRETARIA DE CIÊNCIA,
TECNOLOGIA E INOVAÇÃO



*Conselho Nacional de Desenvolvimento
Científico e Tecnológico*

Referências

- NVIDIA Corporation, "O que é CUDA". Disponível em: http://www.nvidia.com.br/object/what_is_cuda_new_br.html, Março, (2012)
- NVIDIA Corporation, "NVIDIA CUDA C Programming Guide 3.1." Disponível em: http://developer.nvidia.com/object/cuda_download.html, Março, (2011).
- RAVIART P. A. e THOMAS J. M. s, A Mixed Finite Element Method for Second Order Elliptic Problems, vol. 606 of Springer Lecture Notes in Mathematics, New York (1977).
- Goddeke D, Strzodk R, Mohd-Yusof J, McCormick P, H.M Buijssen S, Grajewski M. e Turek S, "Exploring weak scalability for FEM calculations on a GPU-enhanced cluster", (2007).
- Aiping Ding, Tianyu Liu, Chao Liang, Wei Ji, and X George Xu (2011) "EVALUATION OF SPEEDUP OF MONTE CARLO CALCULATIONS OF SIMPLE REACTOR PHYSICS PROBLEMS CODED FOR THE GPU/CUDA ENVIRONMENT".
- Alonso, P., Cortina, R, Martínez-Zaldívar, F. J., Ranilla, J. (2009) "Neville elimination on multi- and many-core systems: OpenMP, MPI and CUDA, J. Supercomputing", in press, doi:10.1007/s11227-009-0360-z, SpringerLink Online Date: Nov. 18.
- GPGPU (2012). Disponível em: <http://gpgpu.org/>, março (2012)

Dúvidas

