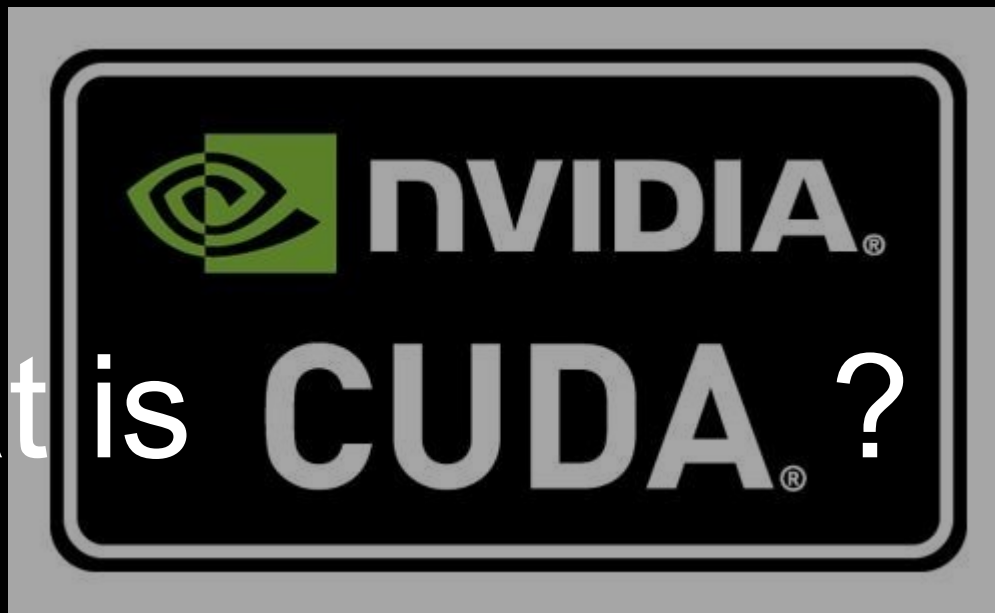


What is **CUDA**?



Eduardo Viola Nicola
evnicola@inf.ufpel.edu.br

Disciplina de IPPD



Sumário

- 1) Introdução
- 2) Princípio Geral de Funcionamento
- 3) Exemplos de Aplicações
- 4) Modelo de Programação
- 5) Linguagens Suportadas
- 6) Exemplos de Produtos no Mercado

Compute Unified Device Architecture

1 - Introdução

- CUDA™ é uma plataforma para processamento paralelo que obtém ganhos de performance utilizando a GPU para executar instruções como se fosse uma CPU
- Criada pela NVYDIA
- Foi projetada para ser usada em aplicações gráficas... hoje em dia não possui apenas essa finalidade
- Se utiliza das GPUs, que foram projetadas para cálculos em ponto flutuante, ou seja, estas operações são executadas mais eficientemente do que na CPU
- Com CUDA é possível alterar algoritmos ou bibliotecas para se beneficiar do paralelismo. Podem ser usadas as linguagens C, C++, C#, Fortran, Java, Python, etc.

2 - Princípio Geral de Funcionamento

Enquanto a CPU possui de quatro a oito núcleos, a GPU consiste em centenas de núcleos menores. Juntos, eles trabalham para "devorar" os dados do aplicativo. Essa arquitetura massivamente paralela é o que traz a computação de alta performance.

2 - Princípio Geral de Funcionamento

Usando linguagens de alto nível, aplicativos executam partes sequenciais de suas cargas de trabalho na CPU – que é otimizada para performance com um único segmento (thread) – ao mesmo tempo em que ocorre um processamento em paralelo na GPU. Todo o processo é acelerado graças ao "processador adicional" GPU.

Esse tipo de computação é possível porque as placas de vídeo de hoje fazem muito mais do que processar imagens: elas processam tarefas de aplicativos projetados para tudo, desde finanças, até medicina.

2 - Princípio Geral de Funcionamento

Compute Unified Device Architecture

2 - Princípio Geral de Funcionamento

Compute Unified Device Architecture

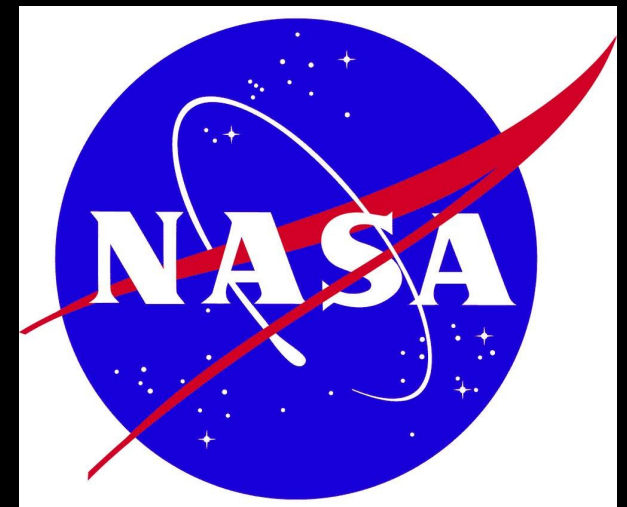
3 - Exemplos de Aplicações

Usar GPUs com o objetivo de simular o fluxo sanguíneo e identificar placas arteriais ocultas sem fazer uso de técnicas invasivas ou cirurgias exploratórias.



3 - Exemplos de Aplicações

- O National Airspace System gerencia o fluxo de tráfego aéreo usando CUDA.
- Equipes da NASA reduziram o tempo de análise do espaço aéreo de dez minutos para três segundos.



3 - Exemplos de Aplicações

Esta empresa resolveu aplicar a CUDA para descobrir qual é a melhor forma de assar uma pizza congelada no forno micro-ondas. Não assou milhares de pizzas, usou modelos virtuais do alimento para descobrir como elas recebiam os efeitos da radiação de micro-ondas.



3 - Exemplos de Aplicações

Aumentar as chances de encontrar petróleo. Como a perfuração do solo custa milhões de dólares, a empresa não pode errar. Por isso, um software de processamento de imagem capaz de analisar dados sísmicos ajuda a organização a encontrar o local correto mais rapidamente.

The logo for SeismicCity features the company name in a bold, blue, sans-serif font. The word "Seismic" is on the left and "City" is on the right. A thin, grey, curved line arches over the "i" in "Seismic" and under the "y" in "City", connecting the two words.

3 - Exemplos de Aplicações

```
__global__ void kernel(float* odata, int height, int width)
{
    unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;
    unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;
    if (x < width && y < height) {
        float c = tex2D(tex, x, y);
        odata[y*width+x] = c;
    }
}
```

Modelo de Programação

4 – Modelo de Programação

- CUDA é um modelo de programação paralela e um ambiente de software projetado para superar este desafio.
- CUDA é composta de três abstrações principais
 - hierarquia de grupo de threads
 - memórias compartilhadas
 - sincronização via barreiras - que são expostos para o programador através de um conjunto mínimo de extensões para a linguagem C

4 – Modelo de Programação

- Estas abstrações possibilitam o paralelismo de dados e threads
 - Particionar o problema em subproblemas
 - Solucionar problemas independentemente em paralelo
 - Combinar as partes para que possam ser trabalhadas cooperativamente

5 - Linguagens Suportadas

C, C++, Fortran

5 - Linguagens Suportadas

- C/C++ → 'CUDA C/C++', compilado com "nvcc", compilador NVIDIA's LLVM-based C/C++
- Fortran → 'CUDA Fortran', usando o compilador PGI CUDA Fortran do The Portland Group.
- Plataforma CUDA suporta outras interfaces computacionais
 - Khronos Group's OpenCL
 - Microsoft's DirectCompute
 - C++ AMP

5 - Linguagens Suportadas

- “Third party wrappers” são disponibilizados:
 - Python
 - Perl
 - Fortran
 - Java
 - Ruby
 - Lua
 - Haskell
 - MATLAB
 - IDL
 - InMathematica (suporte nativo)

Produtos no Mercado

6 – Produtos no Mercado

- Placa Prof Nvidia Quadro K4000 3gb Ddr5
192bits 768cuda,hdmi

6 – Produtos no Mercado



6 – Produtos no Mercado



6 – Produtos no Mercado



6 – Produtos no Mercado

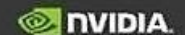


6 – Produtos no Mercado

	Quadro K5000	Quadro K4000	Quadro K2000/D	Quadro K600	Quadro Q410
		<i>New!</i> 	<i>New!</i> 	<i>New!</i> 	
# CUDA Cores	1536	768	384	192	192
Single Precision	2.166 TFLOPs	1.246 TFLOPs	733 GFLOPs	336 GFLOPs	271 GFLOPs
ViewPerf 11	66	53	41	26	19
Memory Size	4 GB + ECC	3 GB	2 GB	1 GB	512 MB
Memory BW	173 GB/s	134 GB/s	64 GB/s	29 GB/s	14 GB/s
Slots + Display Connectors	 2x DP + 2x DVI	 2x DP + DVI	 2x DP + DVI Mini-DP + 2x DVI *	 DP + DVI	 DP + DVI
Display Support**	4	4	4	2	2
Board Power	122 W	80 W	51 W	41 W	40 W

variant of the K2000 - dual DVI + mini-DP sku, available at select OEMs and in the channel.

** DisplayPort 1.2 multi-streaming can be used to drive multiple displays from a single DP connector



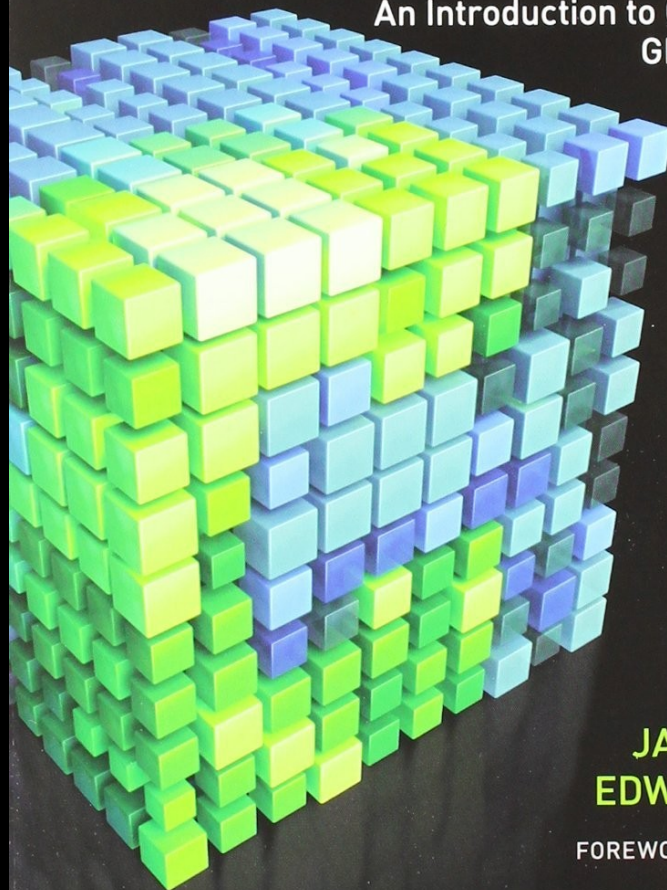
R\$ 2.999,00



CUDA

BY EXAMPLE

An Introduction to General-Purpose
GPU Programming



JASON SANDERS
EDWARD KANDROT

FOREWORD BY JACK DONGARRA

CUDA by Example: An Introduction to General-Purpose GPU Programming

\$26.48 no site Amazon