

# Computação Paralela com GPU - Aula 4 - CUDA

## Agenda do Curso

- Aula 1:
  - Introducao (Nivelamento);
  - Threads;
  - Exemplos Threads.
- Aula 2:
  - OpenMP introducao;
  - OpenMP exemplos;
- Aula 3:
  - OpenCL Introducao;
  - OpenCL exemplos;
- Aula 4:
  - CUDA Introducao;
  - CUDA exemplos;

# Agenda

- Introdução histórica Nvidia
  - Como Nvidia Começou
  - Nascimento CUDA
- O que é CUDA
  - Conceitos Importantes;
  - Execução de Código;
- Porque CUDA
  - Vantagens;
- Exemplos
  - Exemplos CUDA;

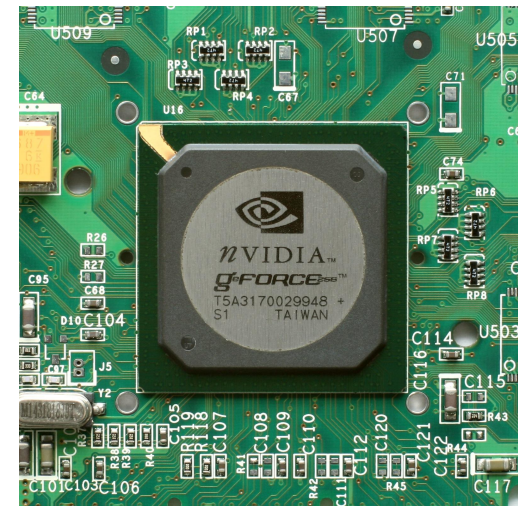
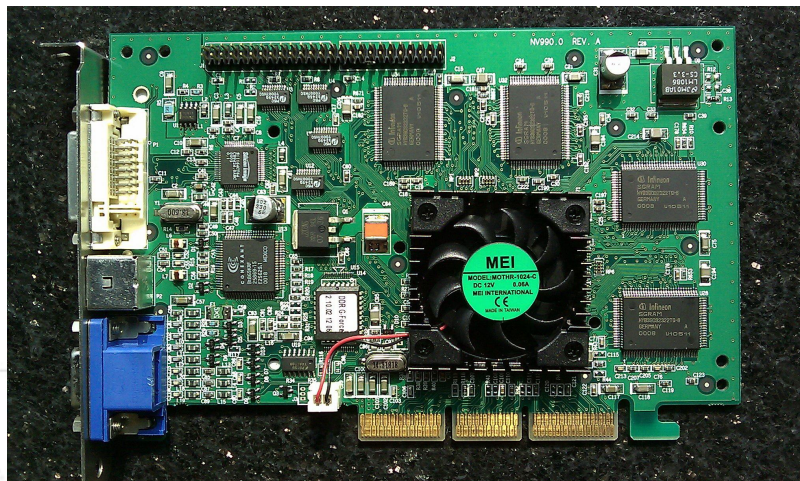
1.

# Introdução

Nvidia , como começou...

# NVIDIA

- CEO e Fundador Taiwanês
  - Microchip designer na AMD
- Ano 1993
- Primeira placa NV10 em 1999
- Uma das pioneiras em hardware de processamento 3d para usuários domésticos.
- Nvidia faz o núcleo da placa de vídeo, existem outras empresas que projetam o resto.





# Streaming processor

- Pensado na Stanford University
- Mudou a arquitetura de GPUs
- Stream Processing
  - Usado em GPU e FPGA;
  - Mais rápido;
  - Não se preocupa com sincronismo;



STANFORD  
UNIVERSITY



## Daí vem CUDA...

- Compute Unified Device Architecture
  - Dá acesso direto ao conjunto de instruções virtuais da GPU;
  - Dá acesso aos elementos de computação paralela.
- Ano 2006;
- Baseada e oferece suporte ao OpenCl.

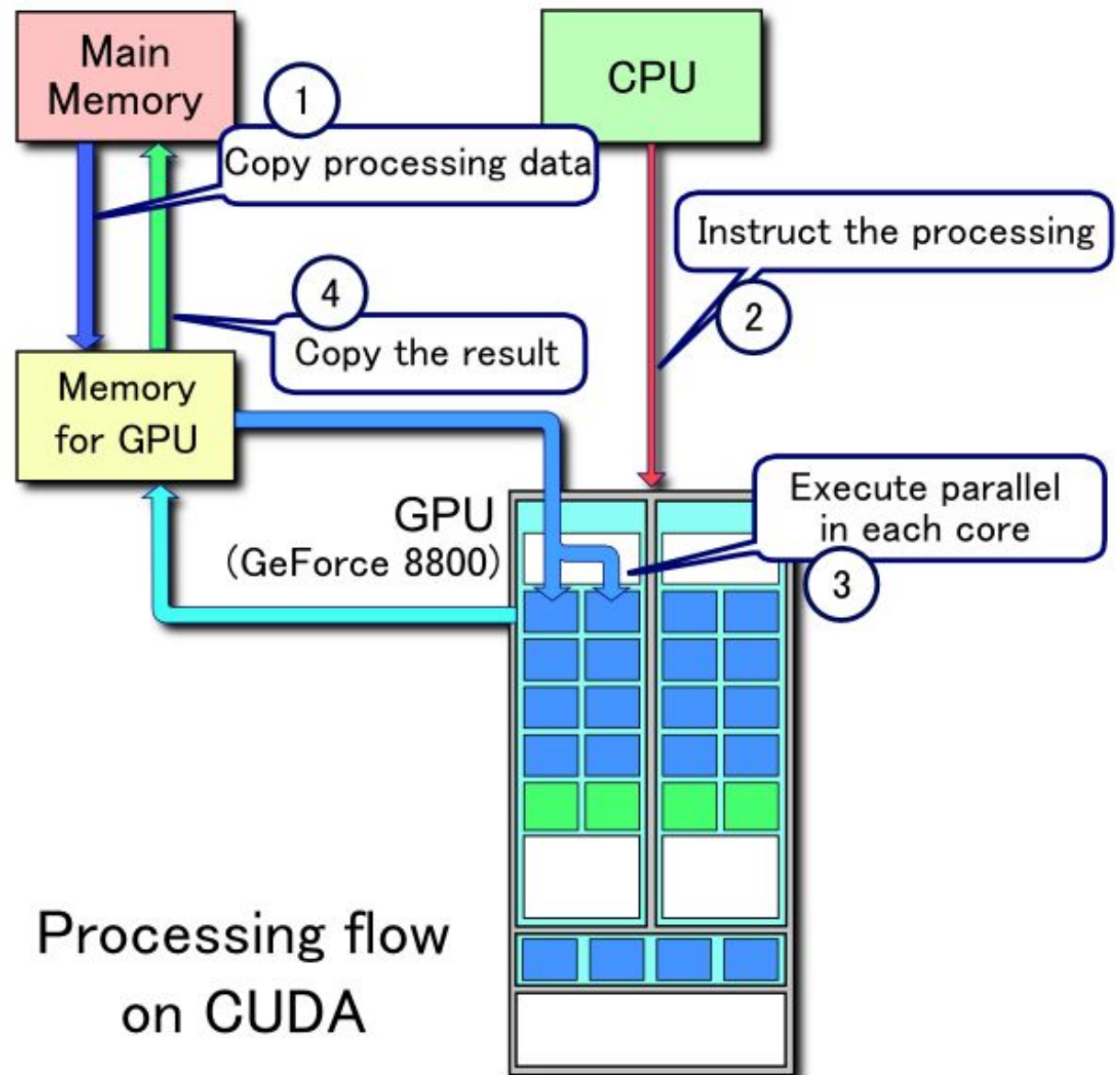
2.

# O que é CUDA



# Conceitos Gerais

- Arquitetura
  - Expõe o paralelismo da GPU para computação de propósito geral;
  - Mantém Performance;
- C/C++
  - Baseado no padrão C;
  - Pequenas modificações possibilitam computação heterogênea;
  - API fácil de usar e intuitiva.



Processing flow  
on CUDA

# Execução do Processo

- Execução de kernel
  - `meukernel<<<1,1>>>();`
  -
- Gerenciar Memória da GPU
  - `cudaMalloc();`
  - `cudaFree();`
  - `cudaMemcpy();`
- Gerenciar comunicação e sincronização
  -

# Execução do Processo

- Execução de kernel
  - `meukernel<<<1,1>>>();`
- Gerenciar Memória da GPU
  - `cudaMalloc();`
  - `cudaFree();`
  - `cudaMemcpy();`
- Gerenciar comunicação e sincronização
  - Barreira : `__syncthreads();`

# Fluxo de Execução

- Inicia Variáveis (Host e Device);
- Aloca espaço (Host e Device);
- Copia para o device;
- Roda o kernel;
- Copia os dados de volta para o host;
- Limpa a memória;



## Tópicos importantes

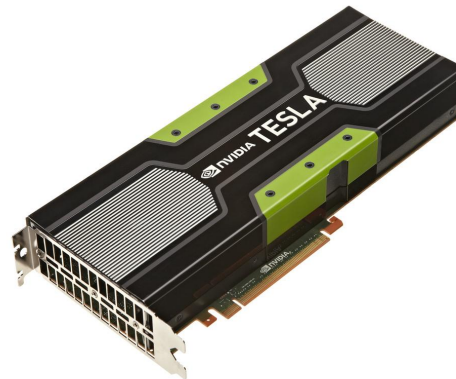
- Computação Heterogênea;
- Blocos;
- Threads;
- Indexação;
- Memória Compartilhada;
- `__syncthread()`;
- Operação Assíncrona;
- Processando Erros;
- Gerenciamento de dispositivos;

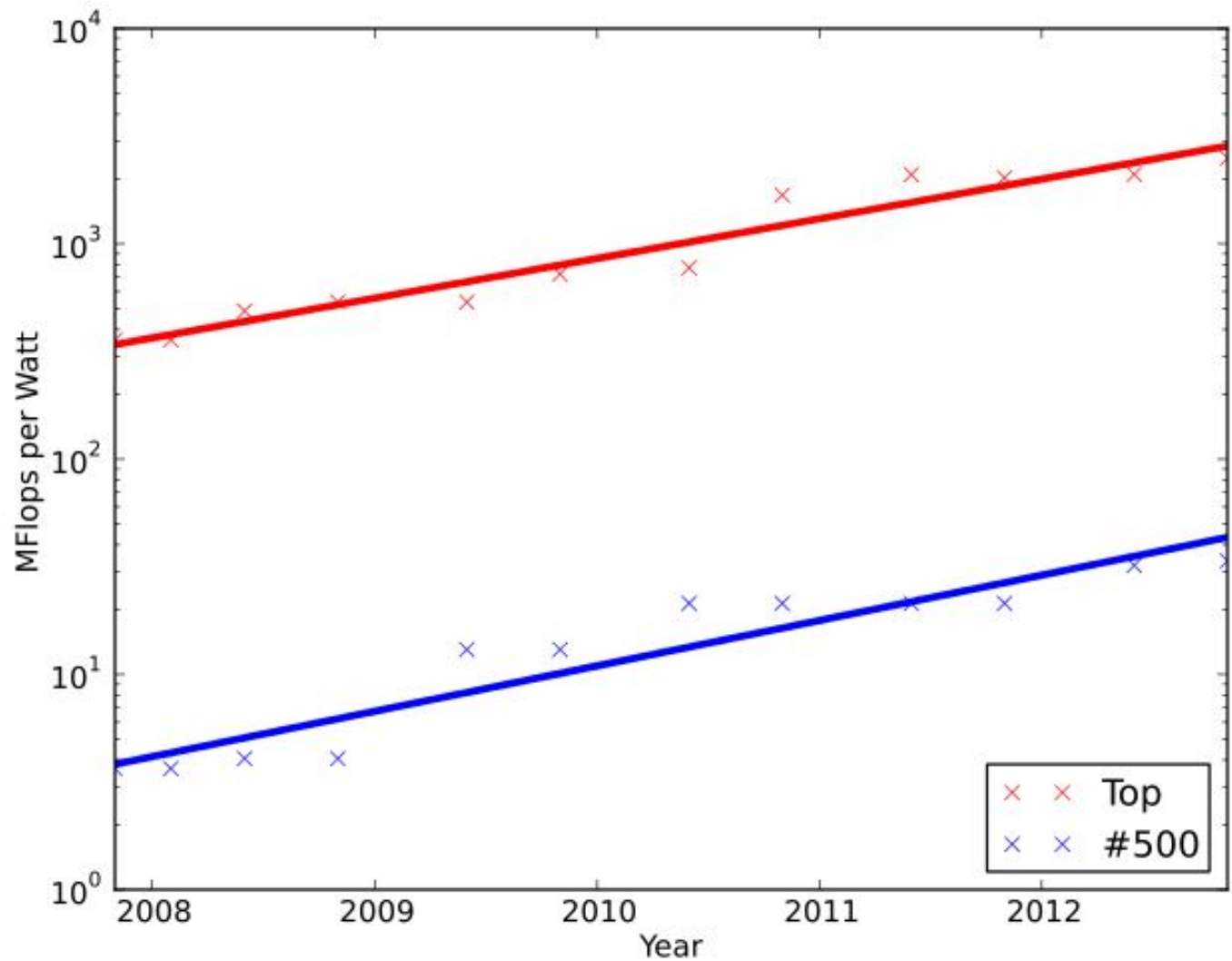
# 3.

## Porque CUDA

## Agenda

- Energéticamente eficiente
  - Performance por watt;
- Tesla K20X VS Xeon
  - Em média 70% mas rápido.





4.

## Exemplos

Alguns exemplos simples...



“

git clone [https://github.com/gaburiero/jornada\\_parallel.git](https://github.com/gaburiero/jornada_parallel.git)

# Exemplos

## 1. CUDA samples

# Bibliografia

- Wikipédia;
- CUDA Developer Zone:
  - <https://developer.nvidia.com/cuda-education-training>
  - <https://developer.nvidia.com/how-to-cuda-c-cpp>
- CUDA by Example: An Introduction to General-Purpose GPU Programming:
  - Book by Edward Kandrot and Jason Sanders

# Obrigado!

## Dúvidas?

Meu contato:

[gabriel.carvalho@itec.ufpa.br](mailto:gabriel.carvalho@itec.ufpa.br)