

Comparação de Eficiência entre as plataformas OpenCL e CUDA em GPUs NVIDIA

Thiago de Gouveia Nunes
Supervisor: Prof. Doutor Marcel P. Jackowski

10 de agosto de 2012

Sumário

1	Introdução	3
1.1	Motivação	3
1.2	Objetivos	3
1.3	Problemas a serem resolvidos	3
2	Conceitos e Tecnologias	4
2.1	High-Performance Computability	4
2.2	GPGPU	4
2.2.1	História	4
2.2.2	Placas NVidia	4
2.3	CUDA	4
2.3.1	Modelo de Memória	4
2.3.2	Modelo de Execução	4
2.3.3	Modelo de Plataforma	4
2.3.4	Modelo de Programação	4
2.4	OpenCL	4
2.4.1	Modelo de Memória	4
2.4.2	Modelo de Execução	4
2.4.3	Modelo de Plataforma	4
2.4.4	Modelo de Programação	4
3	Atividades Realizadas	5
3.1	Comparação de eficiencia	5
3.1.1	Como fazer a comparação?	5
3.1.2	Montagem dos kernels	5
3.2	Comparação das abstrações	5
3.3	Comparação entre os arquivos .ptx	5
4	Resultados	6
4.1	Comparação de eficiencia	6
4.1.1	Kernel memory-bound	6
4.1.2	Kernel processing-bound	6
4.2	Comparação das abstrações	6
4.2.1	Semelhanças	6
4.2.2	Diferenças	6
4.3	Comparação dos .ptx	6
5	Conclusões	7
6	Bibliografia	8
7		9

1 Introdução

1.1 Motivação

GPGPU (general-purpose computing on graphics processing units), ou seja, usar a GPU para realizar computações altamente paralelizáveis, é uma técnica muito usada em computação de alta performance. OpenCL (Open Computing Language) e CUDA (Compute Unified Device Architecture) são linguagens para programas GPGPU.

1.2 Objetivos

O objetivo do estudo é comparar a eficiência dessas duas linguagens rodando em uma placa NVidia GeForce GTX 260.

1.3 Problemas a serem resolvidos

Para realizar essa comparação de eficiência, devemos entender como as linguagens funcionam, suas semelhanças e diferenças, e criar um método que seja justo para comparar programas semelhantes escritos nas duas linguagens.

2 Conceitos e Tecnologias

2.1 High-Performance Computability

HPC nasceu da necessidade de poder computacional para resolver uma série de problemas, entre eles:

- Previsão climática
- Modelação molecular
- Simulações físicas
- Física quântica

Os supercomputadores foram criados para rodar as aplicações que executavam esses objetivos. Até o final dos anos 90 todos os supercomputadores tinham como base CPUs. Só no final da década seguinte, com o aumento do desempenho das GPUs, que alguns supercomputadores começaram a usar GPUs como seus processadores

2.2 GPGPU

2.2.1 História

2.2.2 Placas NVidia

2.3 CUDA

2.3.1 Modelo de Memória

2.3.2 Modelo de Execução

2.3.3 Modelo de Plataforma

2.3.4 Modelo de Programação

2.4 OpenCL

2.4.1 Modelo de Memória

2.4.2 Modelo de Execução

2.4.3 Modelo de Plataforma

2.4.4 Modelo de Programação

3 Atividades Realizadas

3.1 Comparação de eficiencia

3.1.1 Como fazer a comparação?

3.1.2 Montagem dos kernels

3.2 Comparação das abstrações

3.3 Comparação entre os arquivos .ptx

4 Resultados

4.1 Comparação de eficiencia

4.1.1 Kernel memory-bound

4.1.2 Kernel processing-bound

4.2 Comparação das abstrações

4.2.1 Semelhanças

4.2.2 Diferenças

4.3 Comparação dos .ptx

5 Conclusões

6 Bibliografia

