



XXIV
Mostra
de Iniciação
Científica

SEMANA DO
CONHECIMENTO

A Universidade em movimento

De **7 a 10** de outubro de 2014



RESUMO

Análise de ferramentas computacionais voltadas a programação em GPUs

AUTOR PRINCIPAL:

Paulo Souza Junior

E-MAIL:

paulosouzjunior@gmail.com

TRABALHO VINCULADO À BOLSA DE IC::

Pibic UPF ou outras IES

CO-AUTORES:

Prof. Dr. Carlos Amaral Hölbig

ORIENTADOR:

Prof. Dr. Carlos Amaral Hölbig

ÁREA:

Ciências Exatas, da terra e engenharias

ÁREA DO CONHECIMENTO DO CNPQ:

1.03.00.00-7 Ciência da Computação

UNIVERSIDADE:

Universidade de Passo Fundo

INTRODUÇÃO:

A Unidade de processamento gráfico foi desenvolvida para o processamento de gráficos em jogos ou apenas na manipulação de imagens, possuindo diversos processadores e uma estrutura de processamento paralelo. A um tempo atrás programar a GPU para outra finalidade era praticamente impossível. Porém a GPU vem evoluindo, a sua capacidade de processamento vem aumentando significativamente e a sua programação também evoluiu, com o desenvolvimento de frameworks específicos como, as plataformas CUDA, OpenCL e pacotes para o ambiente estatístico R que tem como objetivo tirar proveito da GPU, disponibilizando funcionalidades que permitem adaptar aplicações que antes eram executadas apenas pela CPU. A grande diferença entre a GPU e a CPU é a quantidade de núcleos e a capacidade extremamente diferente de processamento paralelo, devido ao alto número de núcleos na GPU. Com a possibilidade de explorar o poder da GPU busca-se alcançar vantagens na execução desses algoritmos.

METODOLOGIA:

Foram explorados, por base de pesquisa. frameworks específicos desenvolvidos para a implementação de algoritmos que fazem uso de rotinas para utilizar a GPU bem como foram desenvolvidos algoritmos sequenciais e paralelos para coletar resultados do desempenho da CPU. Em um primeiro momento foi estudado a linguagem de programação paralela utilizada para desenvolver programas em CUDA, que mostrou se promissor no desenvolvimento de exemplos de multiplicação de matrizes. A biblioteca OpenCL também foi estudada, porém mostrou-se extremamente complexa no desenvolvimento de programas que fazem uso da GPU. E por fim foi estudado os pacotes utilizados no ambiente R que permitem o uso do processamento gráfico, esses pacotes foram estudados a fundo e percebeu-se que os pacotes do R, por exemplo, gputools faz uso de rotinas já implementadas em CUDA e também CUBlas que é o uso da biblioteca Blas em CUDA.

RESULTADOS E DISCUSSÕES:

Os testes que realizados foram desenvolvidos no grupo de pesquisa ComPaDi da Universidade de Passo Fundo. O computador utilizado possui um processador Intel Core i7 920, que opera à frequência de 2.66 Ghz, com 8 MB de cache L2, 8 GB de memória RAM, sistema operacional Ubuntu 14.04 64 bits e placa de vídeo GeForce GTS250 1GB DDR3 ECS. Os softwares utilizados foram a linguagem R (versão 3.0.1 de 64 bits), a IDE RStudio e o pacote usado foi gputools, bem como foram desenvolvidos programas em CUDA e OpenCL, onde em OpenCL não foi possível coletar os resultados que estariam adequados a comparação em relação ao tempo de execução. O teste implementado foi uma multiplicação de matrizes quadradas de ordem 2000, onde os seguintes resultados foram encontrados: em um ambiente de execução sequencial o tempo de execução foi de 1.665 segundos, já utilizando o pacote que faz uso de CUDA para a linguagem R, chamado gputools, foi executada a mesma multiplicação utilizando a placa de vídeo GeForce GTS250 e o tempo de execução encontrado foi de 0.025 segundos. Com esse resultado foi possível perceber a grande diferença do uso da GPU em relação a CPU, demonstrando assim o seu potencial.

CONCLUSÃO:

Tendo em vista o potencial da GPU, por mais que a sua programação esteja complexa a sua capacidade computacional está bem além dos atuais processadores. Com os modelos de simulação apresentando cada vez mais dados é imprescindível encontrar formas de otimizar o seu desempenho. Para tanto, o uso da GPU pode tornar a computação mais rápida.

REFERÊNCIAS BIBLIOGRÁFICAS:

WILT, Nicholas. The CUDA handbook: A Comprehensive Guide to GPU Programming. 1. ed. Addison-Wesley Professional, 2013.

M. Aftab, G. Benedict, M. G. Timothy, F. James and G. Dan. OpenCL: Programming Guide. 1. ed. Addison-Wesley, 2011. ISBN 9780321749642.

SCARPINO. Matthew. OpenCL in Action: How to accelerate graphics and computation. 1. ed. Manning, 2011. ISBN 9781617290176.

ADLER, J. R in a Nutshell. 2. ed. EUA: O'Reilly, 2012.

Assinatura do aluno

Assinatura do orientador