

Desenvolvimento de Aplicativos para Placas Gráficas com Uso de CUDA

Mark Joselli¹

¹PUC-PR

Mark.joselli@pucpr.br

***Abstract.** The GPU is a hardware dedicated to massively parallel processing and it has an intense mathematical processing. Various applications require solving many mathematical problems, making relevant the GPU for processing. The CUDA is a powerful framework to develop algorithms using graphics cards from nVidia, which today are present even on mobile devices.*

In addition, the cost and the learning of this architecture is a big plus for developers, and will be presented during the tutorial as an excellent choice for anyone who wants to develop algorithms using the power of parallelism of GPUs.

***Resumo.** A GPU é um hardware dedicado ao processamento massivamente paralelo e com um intenso processamento matemático. Diversas aplicações exigem a resolução de muitos problemas matemáticos, fazendo relevante o uso da GPU para o seu processamento. O CUDA é um poderoso framework para desenvolver algoritmos usando placas gráficas da nVidia, que hoje em dia estão presentes até mesmo em dispositivos móveis.*

Além disso, o custo e a aprendizagem dessa arquitetura é uma grande vantagem para desenvolvedores, e será apresentado durante o tutorial como uma excelente escolha para qualquer pessoa que queira desenvolver algoritmos usando o poder do paralelismo da GPUs.

1. Informação Geral

A GPU (Unidades de processamento gráfico) são processadores dedicados para o processamento de cálculos gráficos. O desenvolvimento das GPUs programáveis começou uma nova área de pesquisa, permitindo o uso da GPU para processamento de dados não gráficos, este uso ficou chamado como GPGPU (GPU para uso genérico). Essas GPUs devido a sua arquitetura SIMD (Single Instruction, Multiple Data) são altamente otimizada para elevado numero de cálculos.

Este trabalho visará a utilização dessas GPUs para a utilização em ambiente paralelos como clusters de GPUs, processamento distribuído entre GPUs e utilização da GPU para processamento em grade.

Este tutorial tem o objetivo de capacitar o aluno a começar o desenvolvimento de algoritmos utilizando GPU Computing. Portanto, este tutorial apresenta uma abordagem basica, que vai desde o funcionamento ao primeiros passos de como usar a GPU para

processamento de algoritmos. Além disso, o tutorial irá mostrar algumas técnicas sobre como melhor usar o poder da GPU para otimizar algoritmos, aumentando a velocidade quando comparado com modelos tradicionais. Este tutorial também mostra alguns exemplos destas técnicas na prática.

2. Objetivo principal

Apresentar as arquiteturas paralelas, mostrar técnicas para a utilização de forma otimizada, exemplificar com testes de casos.

3. Objetivos secundários

Instalação e configuração do ambiente de desenvolvimento do CUDA nas diversas plataformas. Compilador CUDA, diretivas de compilação e compilação com código de CPU. Análise e desenvolvimento de código em paralelo na GPU.

4. Metodologia

O tutorial será dado na forma de exposição, onde os alunos interagirão com o expositor e estudos de casos serão apresentados.

5. Nível do tutorial:

Iniciante

6. Motivação

GPU Computing vem sendo amplamente aplicada em diversos campos de pesquisa, desde do início das GPUs programáveis. Atualmente, com o CUDA da NVIDIA, DirectComputing da Microsoft e o OpenCL temos bibliotecas que transformaram a GPU em um cluster de multiprocessadores, sendo utilizada em diversas áreas da computação. Dentre os diversos recursos disponíveis nestas bibliotecas, há a integração com as APIs gráficas, um recurso de grande aplicação em jogos e efeitos visuais.

7. Conhecimentos Necessários

Noções de processamento paralelo e conhecimento de programação básica em C ou linguagem similar.

8. Relevância para o ERAD-RJ

Migrar para a computação paralela pode ser muito difícil, e ter uma curva de aprendizado alta. Dessa forma este tutorial pretende ajudar aos alunos darem os primeiros passos no conceito e uso de programação paralela para placas gráficas.

9. tópicos abordados

O minicurso é dividido em 4 partes. Na primeira parte do curso são apresentados os conceitos de programação paralela de alto desempenho, arquitetura e hardware da GPU. Os conceitos apresentados nesta parte do curso tem o objetivo de fazer o aluno construir o conhecimento teórico para uma visão prática. A segunda parte consiste na

apresentação das ferramentas de desenvolvimento, para a instalação e configuração do ambiente. A terceira parte consiste no estudo do conteúdo que envolve o desenvolvimento de programas com estrutura de dados na forma de fluxo, compilação e conhecimento de diretivas de compilação para otimização do código e conhecimento das APIs necessárias a execução de um código de GPU. Por fim, apresenta-se um overview para o entendimento prático dos problemas inerentes a gerencia e hierarquia de memória da GPU.

1. Computação paralela; GPU Computing – Computação paralela aplicada a GPU; Arquitetura da GPU; SLI – (*Scalable Link Interface*); Multiprocessador x processador de fluxo.
2. Apresentação, instalação e configuração das ferramentas de desenvolvimento: CUDA Driver; CUDA Toolkit; CUDA SDK; Conhecendo o compilador CUDA (NVCC)
3. Aprendendo as APIs do CUDA: Alocação e desalocação de memória; Cópia de memória; Inicializando vetor dentro da GPU; Obtendo informações sobre a GPU.
4. Introdução a linguagem CUDA: Meu primeiro programa “*Hello world* em paralelo”: Chamando o kernel.

10. público-alvo

Profissionais, alunos de Pós-Graduação e de Iniciação Científica

11. Duração

3 horas

12. Biografia

Mark Joselli, é CDO (diretor de desenvolvimento) da Nullpointer Tecnologias, empresa de tecnologia, que atua como laboratório de desenvolvimento em Games, mobile, realidade aumentada, simulações e Gpu Computing. Ele é graduado em engenharia elétrica/eletrônica pelo CEFET-RJ, com mestrado em computação pela UFF e doutorando em computação pela UFF.

Mark é atuante no mercado de jogos digitais possuindo diversos artigos em congressos nacionais e internacionais como num dos principais eventos de games do Brasil (SBGames), e criou o primeiro GPGPU game. Mark Joselli também ministra aulas pela faculdade CCAA e atua como pesquisador no MediaLab Uff, realizando pesquisa em games, GPGPU, interfaces e CG.

13. Experiência sobre o tema do tutorial

Mark Joselli tese sua tese de mestrado e doutorado no tema, além de ter mais de diversos artigos científicos publicados no tema. Além disto, foi palestrante no GTC 2012 (segundo maior congresso de GPU do mundo) falando exatamente sobre o tema desse tutorial.