

Aluno:

Bernardo Breder

Professor:

Esteban Clua

Título:

Otimização e Análise de código CUDA para GRID de GPU

Descrição:

O estudo orientado tem como objetivo especificar e codificar uma ferramenta que analisa um código na linguagem CUDA, resultando em um relatório que caracteriza a execução desse algoritmo. Além disso, essa ferramenta também irá otimizar o código em função da placa de vídeo que será executado.

Resultado Esperado:

O resultado esperado pelo trabalho é ter uma métrica que avalia o ganho de recursos e de tempo que a ferramenta proporciona. Para isso, será avaliado a vantagem em desempenho quando um algoritmo escrito na linguagem CUDA é submetido numa placa de vídeo de configuração específica. Além disso, será esperado que a ferramenta gere uma descrição do algoritmo de tal forma que a GRID de GPU possa analisar e escalonar a execução da melhor forma possível.

Critério de avaliação:

O critério de avaliação desse trabalho é verificar se a ferramenta está gerando um ganho de desempenho da execução otimizado do algoritmo CUDA em placas de vídeo de configuração específica. Uma outra avaliação será feita quanto a descrição do algoritmo que a ferramenta realiza quando é submetido um algoritmo na linguagem CUDA.

Cronograma de Atividade:

O cronograma de atividades foi construído baseado no término do período letivo do calendário da UFF. De acordo com o calendário, o período termina no final do mês de junho. Sendo assim, o cronograma será construído de forma a terminar no meio do mês de junho.

Até o final do mês de Abril, será levantado todos os artigos associados ao Estudo Orientado e será marcado os seus principais.

O mês de Maio será reservado para o desenvolvimento da ferramenta descrito pelo Estudo Orientado.

Por fim, até a metade do mês de Junho, será gerado as métricas de avaliação do trabalho resultando em documentos que avaliam a contribuição dessa ferramenta.

Referências:

- [1] Automated Dynamic Analysis of CUDA Programs
- [2] Source-to-source optimization of CUDA C for GPU Accelerated Cardiac Cell Modeling
- [3] Compiler and Runtime Support for Enabling Generalized Reduction Computations on Heterogeneous Parallel Configurations
- [4] Dynamically Managed Data for CPU-GPU Architectures
- [5] Automatic C-to-CUDA Code Generation for Affine Programs
- [6] Automatic tuning of the sparse matrix vector product on GPUs based on the ELLR-T approach
- [7] Automatic Data Layout at Multiple Levels for CUDA
- [8] CUDA-Zero: a framework for porting shared memory GPU applications to multi-GPUs
- [9] FLAP: Tool to generate CUDA code from sequential C code
- [10] GPU-TLS: An Efficient Runtime for Speculative Loop Parallelization on GPUs
- [11] A Highly Efficient Implementation of I/O Functions on GPU
- [12] A Polyhedral Modeling Based Source-to-Source Code Optimization Framework for GPGPU
- [13] Algorithmic Strategies for Optimizing the Parallel Reduction Primitive in CUDA
- [14] Auto-tuning a High-Level Language Targeted to GPU Codes
- [15] CUDA's Mapped Memory to Support I/O Functions on GPU
- [16] CuMAPz: A Tool to Analyze Memory Access Patterns in CUDA
- [17] Data Handling Inefficiencies between CUDA, 3D Rendering, and System Memory
- [18] Data Structure Design for GPU Based Heterogeneous Systems
- [19] Exploration of Automatic Optimization for eUDA Programming
- [20] On Improving the Performance of Multi-threaded CUDA Applications with Concurrent Kernel Execution by Kernel Reordering
- [21] Optimization of Linked List Prefix Computations on Multithreaded GPUs Using CUDA
- [22] Simple Memory Machine Models for GPUs
- [23] Understanding the Impact of CUDA Tuning Techniques for Fermi
- [24] Optimization Principles and Application Performance Evaluation of a Multithreaded GPU Using CUDA
- [25] Implementing Open-Source CUDA Runtime
- [26] Runtime Compilation with NVIDIA Cuda as a Programming Tool
- [27] NVIDIA Tegra K1 Mobile Processor
- [28] NVIDIA Tegra X1
- [29] Tree Structured Analysis on GPU Power Study
- [30] Generating CUDA code at runtime: specializing accelerator code to runtime data
- [31] A Flocking Boids Simulation and Optimization Structure for Mobile Multicore Architectures

Data:

08/04/2015

Assinatura do Orientador: