Comparação de Desempenho entre OpenCL e CUDA

IME - Instituto de Matemática e Estatística

Thiago de Gouveia Nunes

Orientador: Prof. Marcel P. Jackowski

Introdução

GPGPU (General-purpose computing on graphics processing units) é utilizar uma GPU (Graphic Processing Unit) como recurso de processamento para resolver problemas que utilizariam uma CPU.

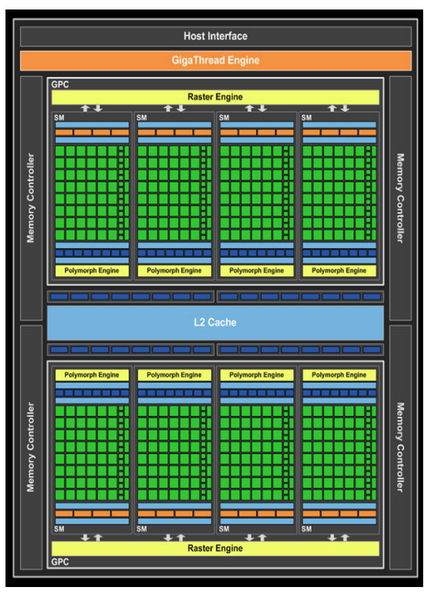
Existem 2 linguagens fortes no mercado para programação GPGPU.
Uma delas é o CUDA, desenvolvida pela NVIDIA para placas NVIDIA, e a outra é o OpenCL,uma alternativa open source desenvolvida por várias empresas em conjunto, e mantida pelo grupo Khronos.

GPGPU nasceu do hardware diferenciado da GPU, que permite que problemas que são altamente paralelizaveis e com o minimo de dependencia entre suas threads sejam executados em uma fração de tempo do que eles seriam numa CPU.

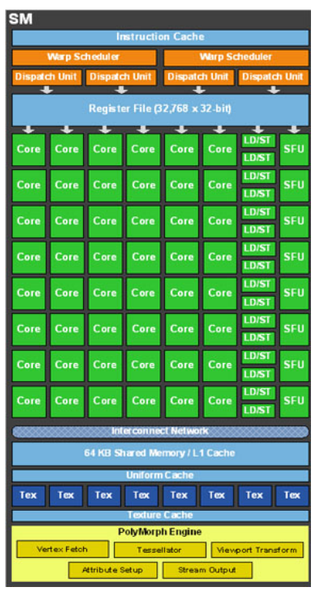
Uma GPU NVIDIA é dividida em vários Streaming Multiprocessors, que
são unidades especiais que agrupam vários processadores, registradores, um bloco de memória compartilhada para seus processadores.

Chamamos de *Kernel* o código que será executado na GPU. Ao lançar um kernel para execução, várias threads são criadas e agrupadas. Cada grupo de threads é escalonado para execução em um SM, e as threads dentro de um SM são agrupadas de 32 em 32 para execução em paralelo.

As threads que estão no mesmo SM executam a mesma instrução ao mesmo tempo, seguindo o paradigma de execução *SIMD* (Single Instruction, Multiple Data).



O hardware da GPU GTX 460 SE



Um SM

Medida de desempenho

Para qualificar as duas linguagens, foram montados 2 tipos de kerneis.

O primeiro usa a GPU para copiar uma matriz de floats para outra. Ele é usada para avaliar a metodologia que as duas linguagens usam para tratar o acesso a memória dentro da GPU.

O segundo kernel faz uma multiplicação entre duas matrizes de floats e guarda o resultado numa terceira. Ele é usado para avaliar o método de execução das duas linguagens.

A GPU usada para os testes é uma GTX 460 SE da NVIDIA.

Resultados dos testes

Para colher resultados dos testes cada um dos kernels rodou 3.000.

O CUDA foi, em média, 15 vezes mais rápido que o OpenCL nos dois kernels.

Os histogramas abaixo mostram o tempo de execução, em milissegundos.

Entendendo os resultados

Uma das causas da diferença de desempenho entre as linguagens é o paradigma usado por elas ao abstrair a execução das threads na GPU.

O OpenCL foi desenvolvido para rodar em sistema hibrídos, em que tanto GPUs e CPUs são usados, e por essa abrangência ele paga com desempenho. Ele não reconhece várias funcionalidades que as placas NVIDIA oferecem, como por exemplo o uso de ponto flutuante de precisão dupla, enquanto o CUDA tem acesso a essas funcionalidades.

A outra causa está na compilação das linguagens para a máquina virtual das GPUs NVIDIA, a PTX (Parallel Thread Execution). Essa máquina virtual é usada para garantir a compatibilidade de um mesmo programa para GPUs diferentes.

É possível compilar os kernels para um arquivo .ptx, que depois será executado nessa máquina virtual.

Ao comparar esses dois arquivos, notou-se que os kernels do CUDA fazem menos acesso a memória global da GPU do que os do OpenCL. Isso faz uma grande diferença, já que essa memória deve ser acessada por todas as threads executando dentro da GPU, gerando um acesso sequêncial para esse acesso, que acaba diminuindo muito a velocidade de execução das threads.