Universidade Federal de Pernambuco

Centro de Informática

Uma Ferramenta MDA para a Plataforma CUDA

Proposta de Trabalho de Graduação

Ademir José de Carvalho Junior

Agosto de 2008

Sumário

1. Introdução

1.1 GPU

GPU (*Graphics Processing Unit*, ou Unidade de Processamento Gráfico) [1], é um tipo de microprocessador especializado no processamento de gráficos. Este dispositivo é encontrado em videogames, computadores pessoais e estações de trabalho, e pode estar situado na placa de vídeo ou integrado diretamente à placa-mãe.

A tarefa de qualquer sistema gráfico 3D é sintetizar uma imagem a partir de uma descrição de cena, 60 vezes por segundo, para gráficos em tempo real como os existentes em videogames. Esta cena contém as primitivas geométricas a serem visualizadas assim como descrições das luzes que iluminam a cena, a maneira como os objetos refletem a luz e a posição e orientação do ponto de visualização.

Esse processo de sintetização ocorre em um *pipeline* de estágios especializados, aonde cada estágio é responsável por uma tarefa, como transformação de coordenadas, cálculos de câmera e iluminação, entre outras tarefas.

A estrutura paralela de processamento das GPUs os torna mais capazes no contexto de processamento gráfico que uma CPU (*Central Processing Unit* ou Unidade Central de Processamento) comum, a ponto de existir uma área de pesquisa denominada GPGPU (*General-Purpose GPU*) [2], que tenta estender esta vantagem para outras áreas de computação de propósito geral.

1.2 CUDA

Com o advento de GPUs e plataformas *multi-core* (múltiplas CPUs), a forma de processamento das aplicações se torna inerentemente paralela. Assim, introduz-se um novo desafio no desenvolvimento de software, que consiste em escalar, de forma transparente, o paralelismo contido em uma aplicação para um número

qualquer de processadores.

Neste contexto, CUDA (*Compute Unified Device Architecture*) [3] é um ambiente de software com um modelo de programação paralela que foi projetado para lidar com este problema. Este ambiente utiliza a linguagem C, objetivando uma baixa curva de aprendizado por parte dos programadores.

No ambiente CUDA, existem três conceitos chaves: uma hierarquia de grupos de *threads*; memórias compartilhadas; e barreiras de sincronização. Estes conceitos são disponibilizados ao programador a partir de um conjunto de extensões à linguagem C, e servem para gerenciar os aspectos de concorrência da plataforma.

1.3 MDA

MDA (*Model-Driven Architecture*) [4] é um padrão definido pela OMG (*Object Management Group*) [5], um consórcio entre empresas de software, indústria, governo, e instituições acadêmicas. O MDA oferece um *framework* conceitual para a definição de um conjunto de padrões que suportam o MDD (*Model-Driven Development*) [6].

Uma abordagem baseada em modelos transfere o foco de desenvolvimento das linguagens de programação de terceira geração para modelos, mais especificamente modelos descritos em UML (*Unified Modeling Language*) [7] e seus *Profiles*. O objetivo dessas abordagens é facilitar o desenvolvimento de aplicações, permitindo o uso de conceitos ligados ao domínio em questão ao invés dos conceitos oferecidos pelas linguagens de programação.

No MDA existem alguns modelos, os principais são o PIM (*Platform Indepedent Model*) e o PSM (*Platform Specific Model*). O PIM é um modelo em que os aspectos da aplicação são descritos sem levar em consideração detalhes da plataforma na qual aplicação está sendo desenvolvida. Já o PSM é um modelo

resultante da adição de detalhes específicos de uma plataforma em um PIM já existente. Esta operação é denominada transformação de modelos e conta com o auxílio de algumas tecnologias como XMI (*XML Metadata Interchange)* [8] e MOF (*Meta-Object Facility*) [9].

Um dos principais desafios do MDD é a dificuldade técnica envolvida na transformação de modelos em código. Existe ainda uma descrença sobre a geração de código eficiente, compacto, e que atenda ao propósito contido em seu modelo. Mas, de acordo com Selic, B. [6], esta é a mesma questão levantada a 40 anos atrás, com a introdução dos compiladores, já que eles também tem a função de transformar objetos descritos em uma linguagem de alto nível para objetos descritos em uma linguagem de mais baixo nível, geralmente linguagem de máquina, e considerando aspectos de performance na transformação.

2. Objetivos

2.1 Objetivos Gerais

Este trabalho de graduação objetiva estudar métodos e técnicas no contexto de MDA, e implementar uma ferramenta que aplique estes conceitos para a produção de aplicações voltadas para a plataforma CUDA.

Além disso, o Trabalho de Graduação proposto neste documento possui como objetivo contribuir com o desenvolvimento de novos trabalhos na área de MDA.

2.2 Objetivos Específicos

Espera-se, ao término do desenvolvimento deste Trabalho de Graduação, uma ferramenta implementada que realiza as seguintes funções:

- Criação e manutenção de modelos que representem aplicações CUDA

- Transformação destes modelos em código CUDA
- Execução destes modelos, possibilitando ao usuário avaliar o seu resultado.

3. Cronograma

3.1 Tabela

Lista de Atividades	Agosto				Setembro				Outubro				Novembro			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Pesquisa Bibliográfica																
Profile UML para CUDA																
Implementação da Ferramenta																
Escrita do Documento																
Revisão do Documento																
Confecção da Apresentação																

3.2 Descrição das Atividades

Pesquisa Bibliográfica: Esta atividade compreende um estudo mais profundo dos conceitos fundamentais deste trabalho, e uma pesquisa por trabalhos e

ferramentas relacionadas que possam contribuir para o desenvolvimento da ferramenta.

Profile UML para CUDA: Esta atividade é responsável pela elaboração de um *Profile* UML para o CUDA, que consiste em identificar as principais abstrações oferecidas pela plataforma e criar elementos UML que representem estas abstrações.

Implementação da Ferramenta: Esta atividade consiste na implementação da ferramenta que irá utilizar o *Profile* UML desenvolvido, para a construção de aplicações para a plataforma CUDA.

Escrita do Documento: Esta atividade consiste na criação do relatório do Trabalho de Graduação. No relatório estarão presentes todos os passos e dificuldades surgidas durante a implementação, bem como o material de pesquisa utilizado durante o trabalho.

Revisão do Documento: Esta atividade representa a etapa final do desenvolvimento do relatório do trabalho, com correções e alterações sugeridas pelos orientadores.

Confecção da Apresentação: Esta atividade, baseada no documento produzido na etapa anterior, consiste na elaboração da apresentação final, responsável por descrever o projeto e os resultados obtidos com este trabalho.

4. Referências

[1] Luebke, D., Humphreys, G. How GPUs Work. *Computer* , vol.40, no.2, pp.96-100, Feb. 2007.

[2] General-Purpose GPU. URL: http://www.gpgpu.org. Visitado em Agosto, 2008.

[3] Compute Unified Device Architecture. URL: http://www.nvidia.com/cuda. Visitado em Agosto, 2008.

[4] OMG Model-Driven Architecture. URL: http://www.omg.org/mda. Visitado em Agosto, 2008.

[5] Object Management Group. URL: http://www.omg.org. Visitado em Agosto, 2008.

[6] Selic, B., "The pragmatics of model-driven development," Software, IEEE , vol.20, no.5, pp. 19-25, Sept.-Oct. 2003.

[7] Rumbaugh, J., Jacobson, I., Booch, G., The Unified Modeling Language Reference Manual, Addison-Wesley, 1998.

[8] Object Management Group. XML Metadata Interchange Specification, Version 1.1. February, 1999.

[9] OMG Meta Object Facility Specification, Version 1.3, September, 1999.

5. Assinaturas

25 de Agosto de 2008

Orientadora:

_____
Judith Kelner

Co-orientador:

_____
Thiago Farias

Aluno:

_____
Ademir José de Carvalho Junior