Pontifícia Universidade Católica
do Rio de Janeiro

Andrei Alhadeff Monteiro

Many-core Fragmentation Simulation

DISSERTAÇÃO DE MESTRADO

Dissertation presented to the Postgraduate Program in Informatics, of the Departamento de Informática, PUC–Rio as partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Waldemar Celes Filho

Rio de Janeiro
August 2011

Pontifícia Universidade Católica
do Rio de Janeiro

Andrei Alhadeff Monteiro

Many-core Fragmentation Simulation

Dissertation presented to the Postgraduate Program in Informatics, of the Departamento de Informática do Centro Técnico Científico da PUC–Rio, as partial fulfillment of the requirements for the degree of Mestre.

Prof. Waldemar Celes Filho
Advisor
Departamento de Informática — PUC–Rio

Prof. Marcelo Gattass
Departamento de Informática — PUC-Rio

Prof. Diego Fernandes Nehab
Instituto Nacional de Matemática Pura e Aplicada (IMPA)

Prof. Ivan Fabio Mota de Menezes
Tecgraf — PUC-Rio

Prof. José Eugenio Leal
Coordinator of the Centro Técnico Científico — PUC–Rio

Rio de Janeiro, August 17, 2011

Andrei Alhadeff Monteiro

Andrei Alhadeff Monteiro graduated from Pontifícia Universidade Católica do Rio de Janeiro in Computer Engineering. He then obtained a Master degree at Pontifícia Universidade Católica do Rio de Janeiro in Computer Science, acting in the areas of physics animation and engineering simulation together with GPU programming. While doing his Masters, he worked as a researcher at Tecgraf PUC-Rio with reservoir simulation and rendering.

Bibliographic data

Monteiro, Andrei Alhadeff

Many-core Fragmentation Simulation / Andrei Alhadeff Monteiro ; advisor: Waldemar Celes Filho. — 2011.
59 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)-Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2011.
Inclui bibliografia

1. Informática – Dissertação. 2. Simulação de fragmentação. 3. Múltiplos processadores. 4. CUDA. 5. Método dos elementos finitos. 6. Elementos coesivos. I. Filho, Waldemar Celes. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

To my family, for all the support they have given throughout my life. To my father, Ivan de Castro Monteiro, my mother, Myriam Alhadeff Monteiro, and my sister, Camila Alhadeff Monteiro.

To my adviser, Waldemar Celes, without whom the research would not be possible. Thank you for motivating me throghout these whole years as my adviser and teacher.

To Tecgraf/PUC-Rio laboratory, for giving me opportunity to face such challenges and learning with them.

To all my friends for their support and friendship.

Resumo

Monteiro, Andrei Alhadeff; Filho, Waldemar Celes. **Implementação de simulação de fragmentação em arquitetura de multi-processadores**. Rio de Janeiro, 2011. 59p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Apresentamos um método computacional na GPU que lida com eventos de fragmentação dinâmica, simulados por meio de elementos de zona coesiva. O trabalho é dividido em duas partes. Na primeira parte, tratamos o pré-processamento de informações e a verificação de corretude e eficácia da inserção dinâmica de elementos coesivos em malhas grandes. Para tal, apresentamos uma simples estrutura de dados topológica composta de triângulos. Na segunda parte, o código explícito de dinâmica é apresentado, que implementa a formulação extrínsica de zona coesiva, onde os elementos sção inseridos dinamicamente quando e onde forem necessários. O principal desafio da implementação na GPU, usando a formulação de zona coesiva extrínsica, é ser capaz de adaptar dinamicamente a malha de uma forma consistente, inserindo elementos coesivos nas facetas fraturadas. Para isso, a estrutura de dados convencional usada no código de elementos finitos (baseado na incidência de elementos) é estendida, armazenando, para cada elemento, referências para elementos adjacentes. Para evitar concorrência ao acessar entidades compartilhadas, uma estratégia convencional de coloração de grafos é adotada. Na fase de pré-processamento, cada nó do grafo (elementos na malha) é associado a uma cor diferente das cores de seus nós adjacentes. Desta maneira, elementos da mesma cor podem ser processados em paralelo sem concorrência. Todos os procedimentos necessários para a inserção de elementos coesivos nas facetas fraturadas e para computar propriedades de nós são feitas por threads associados a triângulos, invocando um kernel por cor. Computações em elementos coesivos existentes também são feitas baseadas nos elementos adjacentes.

Palavras–chave

Simulação de fragmentação; Múltiplos processadores; CUDA; Método dos elementos finitos; Elementos coesivos;

Abstract

Monteiro, Andrei Alhadeff; Waldemar Celes (Advisor). **Many-core Fragmentation Simulation**. Rio de Janeiro, 2011. 59p. MSc Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A GPU-based computational framework is presented to deal with dynamic failure events simulated by means of cohesive zone elements. The work is divided into two parts. In the first part, we deal with pre-processing of the information and verify the effectiveness of dynamic insertion of cohesive elements in large meshes. To this effect, we employ a simplified topological data structured specialized for triangles. In the second part, we present an explicit dynamics code that implements an extrinsic cohesive zone formulation where the elements are inserted on-the-fly, when needed and where needed. The main challenge for implementing a GPU-based computational framework using extrinsic cohesive zone formulation resides on being able to dynamically adapt the mesh in a consistent way, inserting cohesive elements on fractured facets. In order to handle that, we extend the conventional data structure used in finite element code (based on element incidence) and store, for each element, references to the adjacent elements. To avoid concurrency on accessing shared entities, we employ the conventional strategy of graph coloring. In a pre-processing phase, each node of the dual graph (bulk element of the mesh) is assigned a color different to the colors assigned to adjacent nodes. In that way, elements of a same color can be processed in parallel without concurrency. All the procedures needed for the insertion of cohesive elements along fracture facets and for computing node properties are performed by threads assigned to triangles, invoking one kernel per color. Computations on existing cohesive elements are also performed based on adjacent bulk elements.

Keywords

Fragmentation simulation; Many-core; CUDA; Finite element method; Cohesive elements;

Contents

List of Figures

List of Tables

If music be the food of love, play on,
Give me excess of it; that surfeiting,
The appetite may sicken, and so die.

William Shakespeare, *Twelfth Night Act 1.*