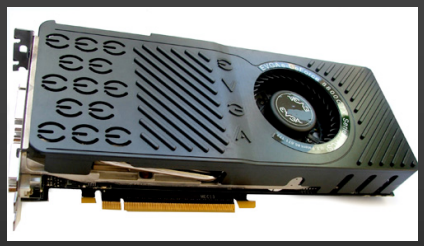
Next Generation - Graphics Hardware

Marcelo Walter
UFPE

atualização/maio 2009

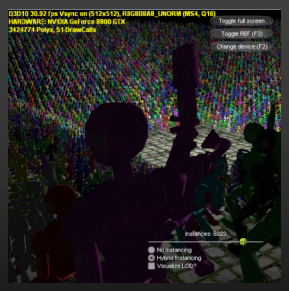
The Sixth Generation (11/2006)

- NVIDIA 8800 series & ATI R600 series
- Novo pipeline gráfico
- DirectX 10



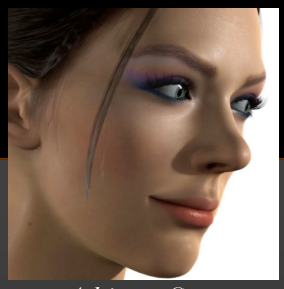
2

Algumas demonstrações



3

Holy Hardware Batman!



Adrianne Curry

- http://www.nvidia.com/docs/CP/45092/techdemo01.swf
- http://www.nvidia.com/docs/CP/45092/techdemo02.swf
- http://www.nvidia.com/docs/CP/45092/techdemo03.swf

Vídeos em Tempo Real…

4

E tudo isto por quanto mesmo?

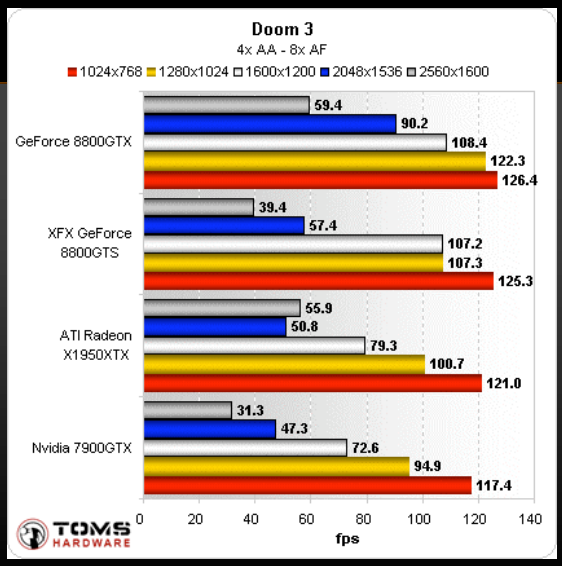
GeForce 8800 GTX Graphics Board

Core	575MHz
Stream Processors	128
Shader	1350MHz
Memory	900MHz
Memory	768MB GDDR3
Outputs	DL-DVI DL-DVI HDTV-out

!!

5

Benchmark



6

Porque todo este “barulho”?

- *Unified Shader*
- *Geometry Shader*
- …e mais umas “coisinhas”…

7

Aplicações não tem um padrão regular de uso…

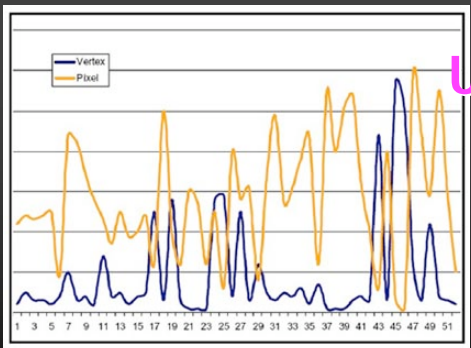


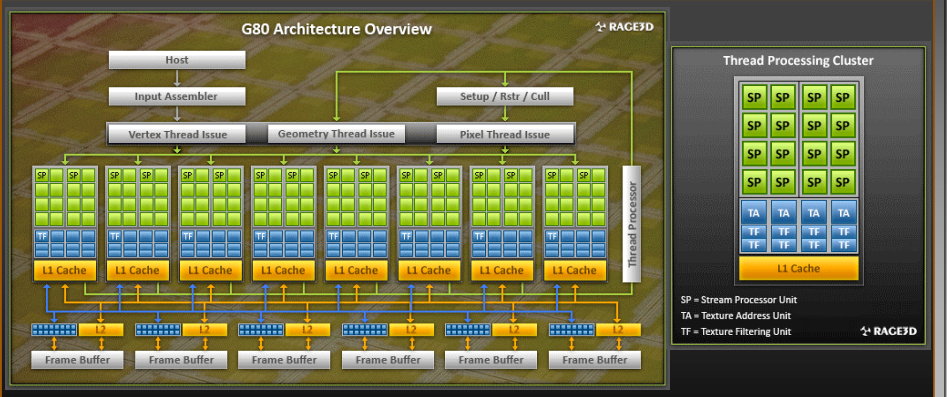
Figure 14. Characteristic Pixel and Vertex Shader Workload Variation Over Time

Unified Shader

No more independent vertex and fragment processors

8

G80 Unified Architecture

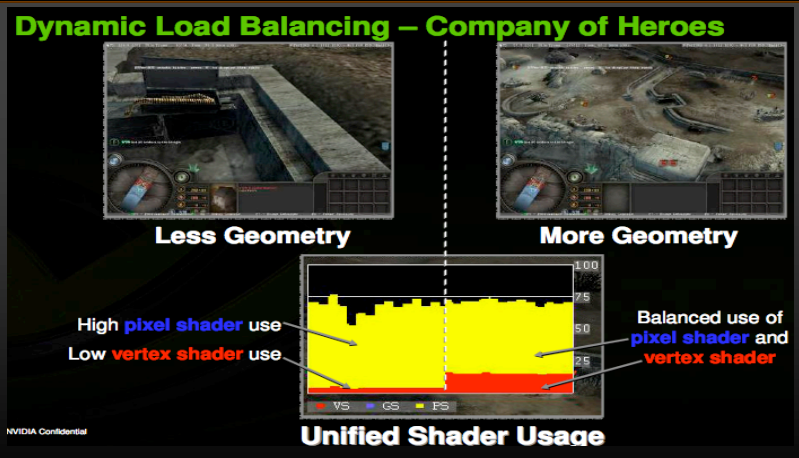


16x8 = 128 Stream Processors (SP)

Generalized floating-point unit that can operate on pixel, vertex, geometry, or even physics operations

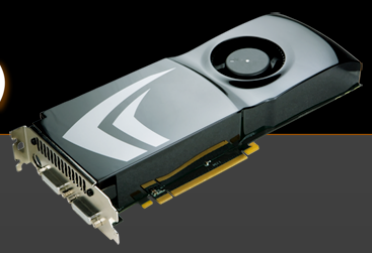
9

Balanceamento de Carga Dinâmico



10

Series 9000
9800 GTX (julho 2008)



Stream Processors	128
Core Clock (MHz)	675 MHz
Shader Clock (MHz)	1688 MHz
Memory Clock (MHz)	1100 MHz
Memory Amount	512MB
Memory Bandwidth (GB/sec)	70.4
Texture Fill Rate (billion/sec)	43.2

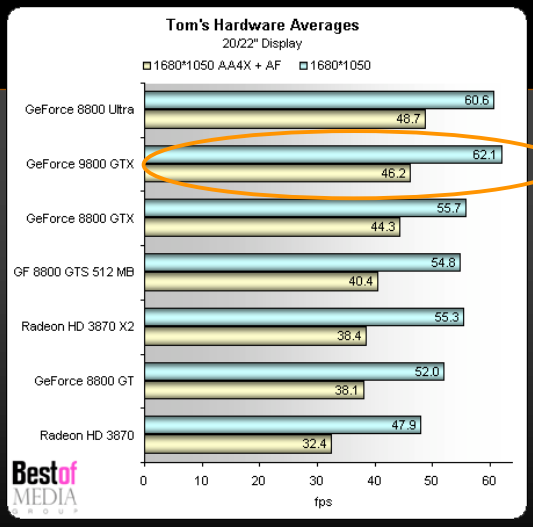
11

Sobre os sufixos da NVidia

- GTX = Enthusiasts
- GTS = Performance
- GT = Mainstream
- G = Entry-level

12

Series 9000
9800 GTX



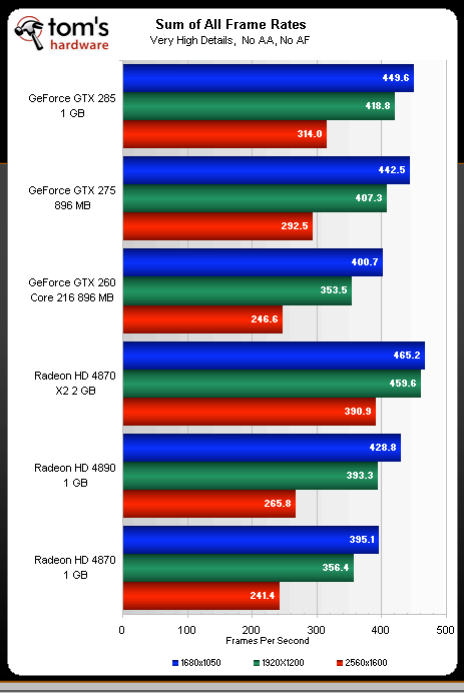
13

Series 200
GTX 275
(Abril 2009)

Stream Processors	240
Core Clock (MHz)	633 MHz
Shader Clock (MHz)	1404 MHz
Memory Clock (MHz)	1134 MHz
Memory Amount	896 MB
Memory Bandwidth (GB/sec)	127
Texture Fill Rate (billion/sec)	50.6

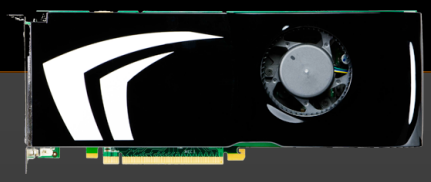
14

Desempenho



15

Series 100
GTS 150
(Março 2009)



- **Não disponíveis para indivíduos**

Stream Processors	128
Core Clock (MHz)	738 MHz
Shader Clock (MHz)	1836 MHz
Memory Clock (MHz)	1000 MHz
Memory Amount	1GB
Memory Bandwidth (GB/sec)	64
Texture Fill Rate (billion/sec)	47.2

16

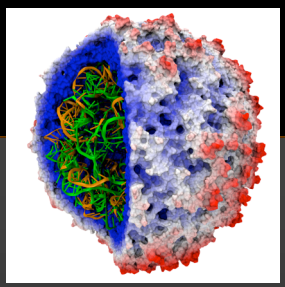
CUDA Anyone?

NVIDIA CUDA ZONE

- ***Compute Unified Device Architecture - 2006 (nome original não mais utilizado)***
- **Arquitetura para Programação Paralela em GPU**
- **C simples**
- **GPUs para as massas!**
- ***Goodies*:**
  - Standard numerical libraries for FFT (Fast Fourier Transform) and BLAS (Basic Linear Algebra Subroutines)
  - Unified hardware and software solution for parallel computing on CUDA-enabled NVIDIA GPUs
  - MathWorks MATLAB® Plug-in

17

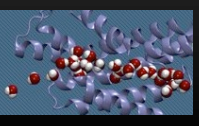
CUDA Anyone?



Exemplos de usos:

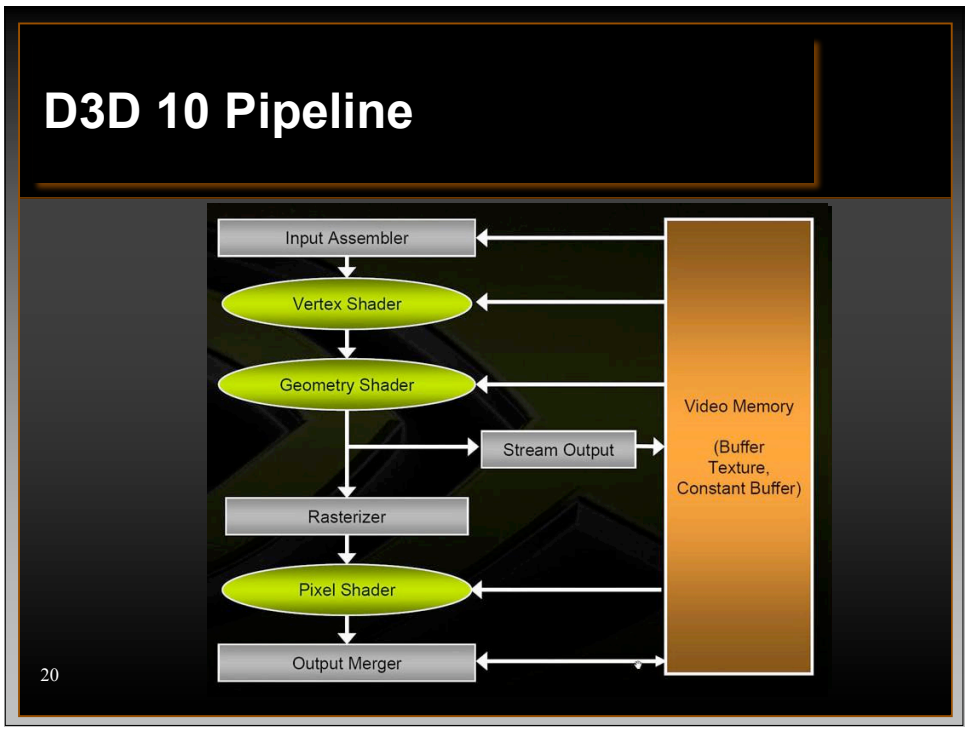
- MDGPU: Molecular Dynamics simulation
- Astrophysical simulations based on smoothed particle hydrodynamics: Fourier Volume Rendering
- Computational biology string matching: CMATCH

THEORETICAL *and* COMPUTATIONAL BIOPHYSICS GROUP
NIH RESOURCE FOR MACROMOLECULAR MODELING AND BIOINFORMATICS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN



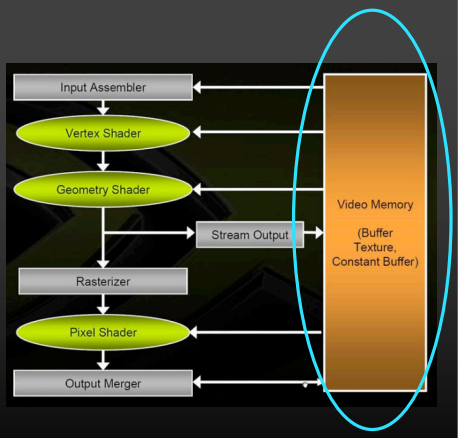
18


The Direct3D
10
Windows Vista


D3D 10 Pipeline
Input Assembler
Vertex Shader
Geometry Shader
Stream Output
Rasterizer
Pixel Shader
Output Merger
Video Memory
(Buffer
Texture,
Constant Buffer)
20

Gerenciamento Unificado de Memória

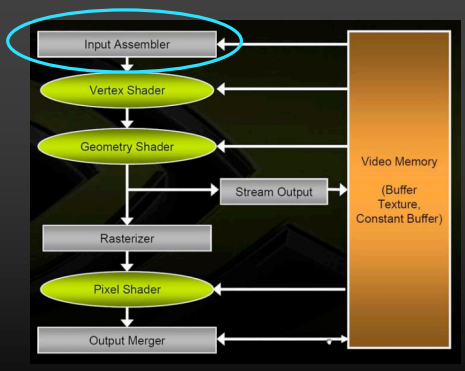
- *Windows Display Driver Model* (WDDM)
- "Tudo" é memória de vídeo



21

Input Assembler

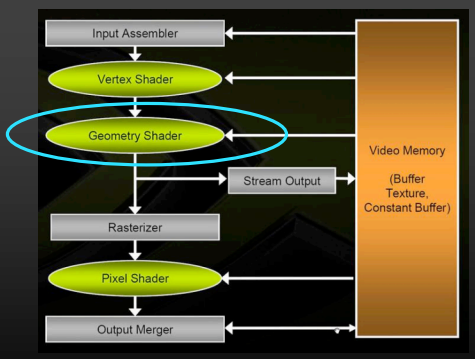
- Lê e monta os dados geométricos
- Adiciona semântica aos dados geométricos
  - *String* que contém informação sobre o uso previsto do parâmetro
- Ex: COLOR, POSITION



22

Geometry Shader

- Permite a GPU **criar** e **destruir** geometria
- Após *Vertex Shader*
- 0-1024 vértices por chamada
- Duas saídas possíveis
  - Rasterizador
  - Stream Output



23

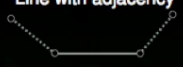
Geometry Shader

Geometry Shader

input

Point
Line Strips
Triangle Strips

Line with adjacency



Triangle with adjacency

output

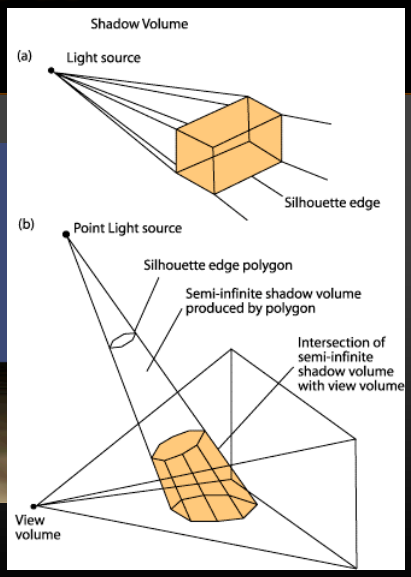
Point list
Line strip
Triangle strip

24

Aplicações do *Geometry Shader*

- *Volume Shadows*
- *Curvas Bézier*
- *Fur generation*
- *Displacement Mapping*
- *Render to cubemap*

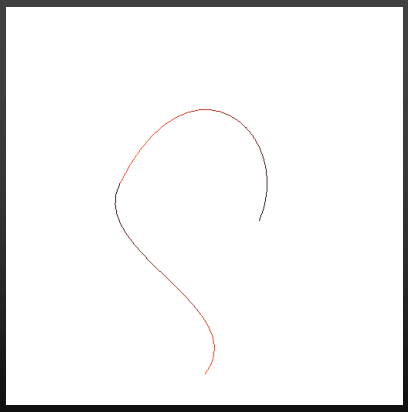
Geração de Volumes de Sombras



26

Geração de Curvas Bézier

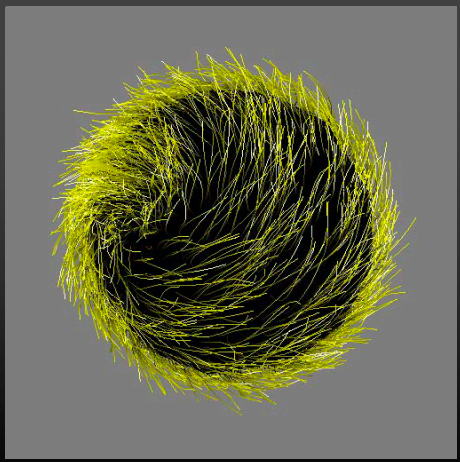
- **Saída**
- **Line Strip**



27

Exemplo: Geração de pêlos

- **Gerar pêlos a partir dos triângulos**
- **1a. Passada**
  - gera linhas aleatórias
- **2a. Passada**
  - gera curvas a partir das linhas (ex. anterior)

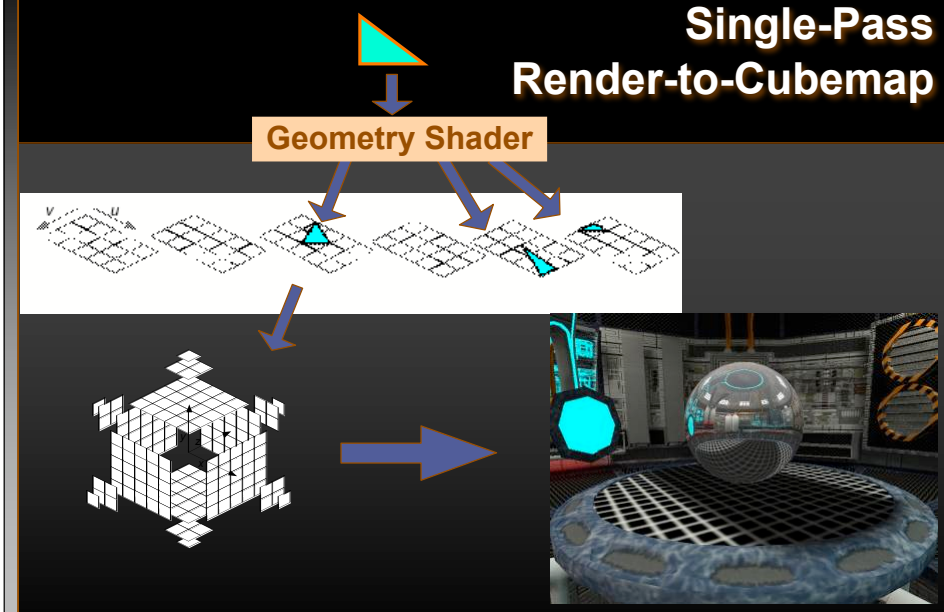


28

Displacement Mapping (D3D 10)

29

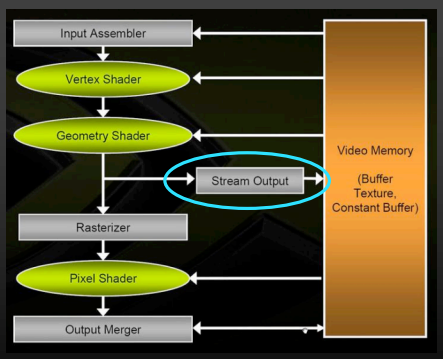
Single-Pass Render-to-Cubemap



30

Stream Output

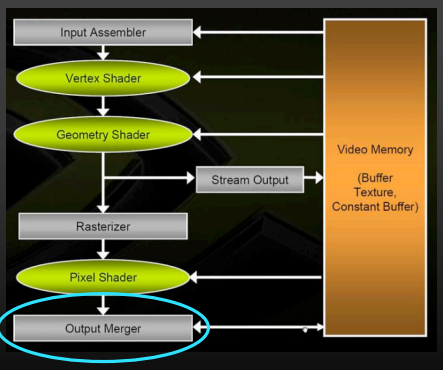
- Geometria gerada no VS/GS pode ser direcionada para um buffer
- Não passa adiante a info de adjacência
- Geometria gerada é **facilmente renderizada** utilizando comando **DrawAuto()** SEM passar pela CPU



31

Output Merger

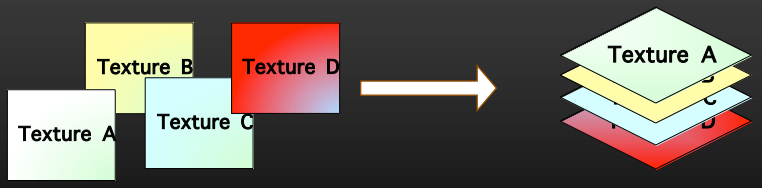
- Combina vários tipos de dados de saída (*pixel shader values, depth and stencil information*) com os conteúdos dos buffers de stencil e profundidade para gerar o resultado final
- Blending: combina o valor de 2 ou mais fragmentos em um de saída



32

Arrays de Texturas

- Colocar as texturas de um mesmo objeto num único array
- Todas com mesma resolução
- Atlas de Textura
- Até 512 elementos no array



33

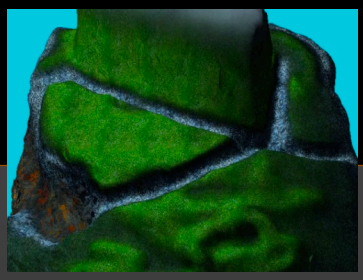
Arrays de Texturas

- Indexados dinamicamente no Shader

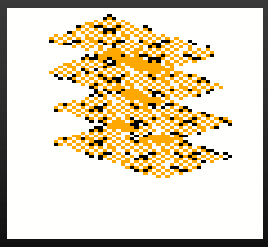
tes da geometria
ndexadas em texturas
diferentes, dentro do
mesmo array

34

Exemplo



Terreno gerado com array de texturas



Geometry Shader

35

Queries & Predicates

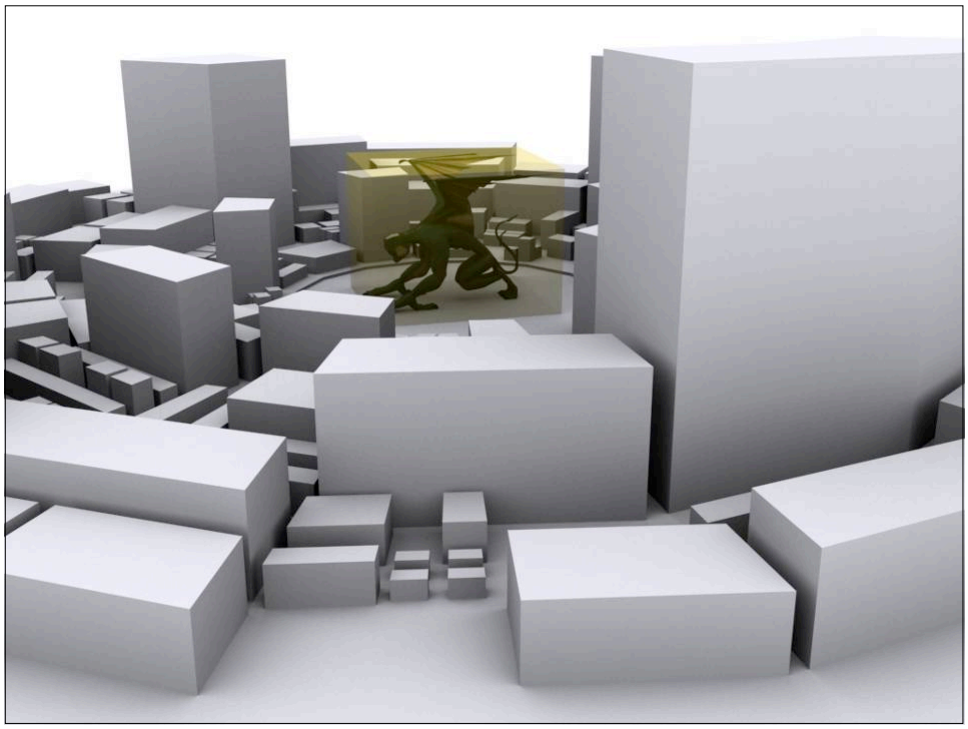
- Commandos podem ser *queued* dependendo do resultado de uma *query*
- Isto é um *Predicate*

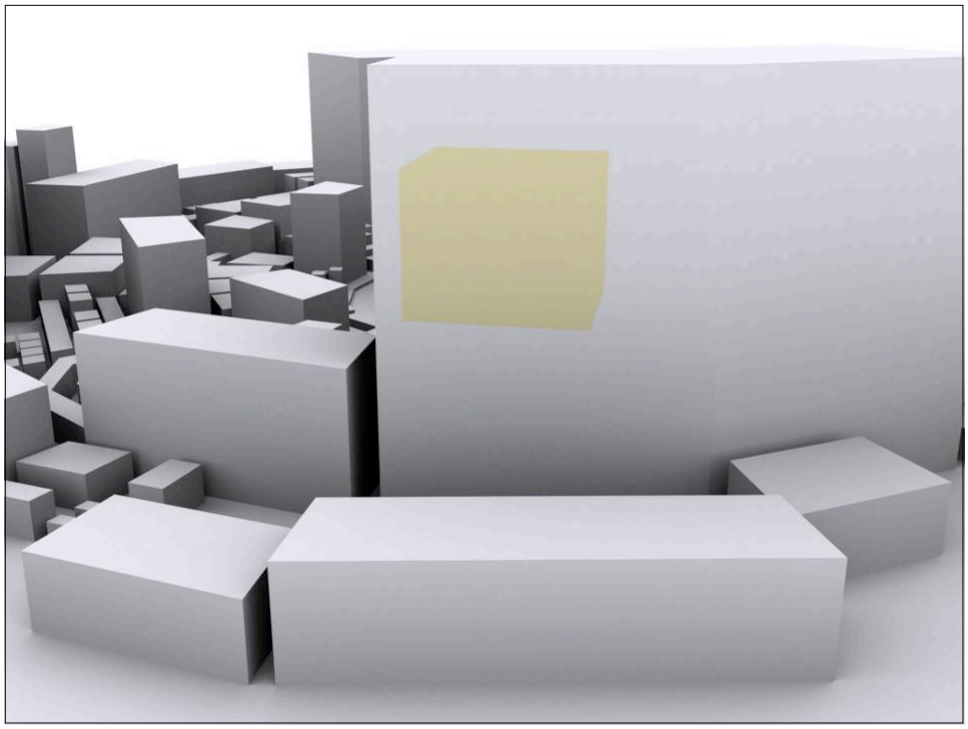
36

Exemplo: *Predicated Rendering*

- **Dependendo de uma *occlusion query* de um *bounding box*, não desenha um objeto**
  - OCCLUSIONPREDICATE
  - Não envolve a CPU

37





Outras novidades

- **Novos formatos Floating Point para HDR**
  - R9G9B5E5_SHAREDEXP
  - R11G11B10_FLOAT

E OpenGL?

- **OpenGL Extensions**
- **GLSL**
  - EXT_gpu_shader4
- **NV_gpu_program4**
  - NV_vertex_program4
  - NV_fragment_program4
  - NV_geometry_program4

NVIDIA

GeForce 8800 OpenGL Extensions

41

Tendências
General Purpose GPUs

GPGPU

General-Purpose Computation Using Graphics Hardware

Home/News | Forums | History | Developer | Supercomputing Course | SC2006 Workshop

Introduction

GPGPU stands for *General-Purpose computation on GPUs*. With the increasing programmability of commodity graphics processing units (GPUs), these chips are capable of performing more than the specific graphics computations for which they were designed. They are now capable coprocessors, and their high speed makes them useful for a variety of applications. The goal of this page is to catalog the current

Graphic-Card Cluster for Astrophysics (GraCCA) -- Performance Tests

Abstract: "In this paper, we describe the architecture and performance of the GraCCA system, a Graphic-Card Cluster for Astrophysics simulations. It consists of 16 nodes, with each node equipped with 2 modern graphic cards, the NVIDIA GeForce 8800 GTX. This computing cluster provides a theoretical performance of 16.2 TFLOPS. To demonstrate its performance in astrophysics computation, we have implemented a parallel direct N-body simulation program with shared time-step algorithm in this system. Our system achieves a measured performance of 7.1 TFLOPS and a parallel efficiency of 90% for simulating a globular cluster of 1024K particles. In comparing with the GRAPE-6A cluster at RIT (Rochester Institute of Technology), the GraCCA system achieves a more than twice higher measured speed and an even higher performance-per-dollar ratio. Moreover, our system can handle up to 320M particles and can serve as a general-purpose computing cluster for a wide range of astrophysics problems. (Hsi-Yu Schive, Chia-Hung Chien, Shing-Kwong Wong, Yu-Chih Tsai, Tzihong Chiueh, Graphic-Card Cluster for Astrophysics (GraCCA) -- Performance Tests. submitted to New Astronomy, 20 July, 2007.)

Posted: 27 Jul 2007 [*GPGPU /Scientific Computing*] #

High Performance Direct Gravitational N-body Simulations on Graphics Processing Units -- II: An implementation in CUDA

Abstract: "We present the results of gravitational direct N-body simulations using the Graphics Processing Unit (GPU) on a commercial NVIDIA GeForce 8800GTX designed for gaming computers. The force

Categories

GPGPU (268)
- Advanced Rendering (32)
  - Global Illumination (13)
  - Image-Based Modeling & Rendering (5)
- Audio and Signal Processing (3)
- Computational Geometry (12)
  - GIS (1)
  - Surfaces and Modeling (3)
- Conferences (15)
- Contests (1)
- Data Parallel Algorithms (3)
- Database (6)
  - Sort & Search (2)
- GPUs (7)
- High-Level Languages (21)
- Image And Volume Processing (36)
  - Compression (1)
  - Computer Vision (6)
- Med & Bio (1)
- Miscellaneous (39)
  - Books (5)
  - Courses (14)
  - Developer Resources (12)
  - Journals (2)
  - Research Groups (2)
- Press (5)
- Scientific Computing (59)
  - Data Compression (2)
    - Data Structures (1)
  - Dynamics Simulation (3)
  - Mathematics (1)
  - Numerical Algorithms (6)
- Site News (7)
- Stream Processing (1)

42

Tendências
Clusters de GPUs?

- *Distributed Texture Memory in a Multi-GPU Environment.* **Graphics Harware 2006**
- *GPU Cluster for High Performance Computing.* **Fan et al.** *ACM / IEEE Supercomputing Conference 2004*
  Cluster de 30 GPUs (GeForce FX 5800 ultra)
  4.6 vezes mais rápido de que cluster de CPUs

43

Tendências

- **GPU/VPU serão muito mais rápidas em termos de clock, mas terão que utilizar tecnologias avançadas de resfriamento (i.e., líquidos).**
- **Em 2010, uma GPU/VPU terá poder de processamento equivalente à um computador que, hoje, estaria nos top 10 do mundo todo.**

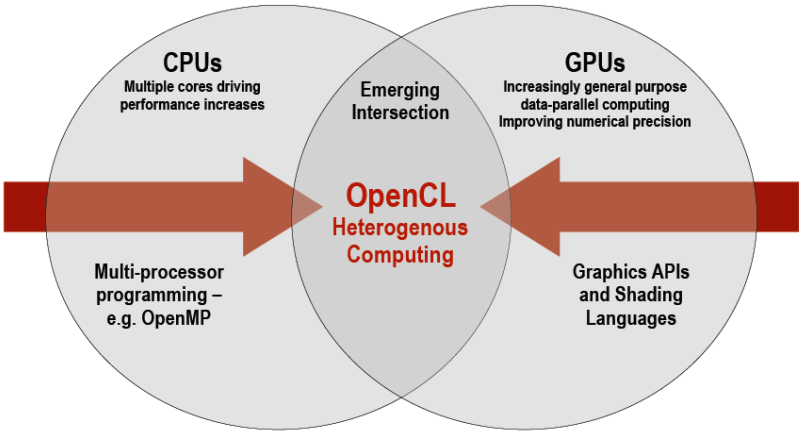
44

Tendências

- OpenCL - Open Computing Language (Dezembro 2008)
- Desenvolvido pelo Khronos group
- Escrita de programas que rodam em ambientes heterogêneos de processadores (CPUs, GPUs)

45

Processor Parallelism



46

Tendências

- GDC 2009 - AMD e Havok demonstram simulações de tecidos escritas em OpenCL
- Abril 2009 - Nvidia lança OpenCL Early Access Program
- Roda em CUDA

47

Desafios para o futuro

- *"It is very difficult to make an accurate prediction, especially about the future"*

Niels Bohr

48

Desafios para o futuro

- **Projeto de uma arquitetura para iluminação global**
- **Num contexto de Ray Tracing, por exemplo:**
  - Teste rápido de intersecção entre superfícies curvas
  - Busca rápida em estruturas de dados espaciais
  - Gerência de cenas muito grandes
    - Em renderização scanline um triângulo pode ser descartado após ser processado
    - Em Ray Tracing isto não é possível, pois há necessidade da informação geométrica próxima para os cálculos globais

49

Desafios para o futuro

- **Projeto de arquiteturas muito pequenas com recursos escassos**
  - Pouca área de chip
  - Pouca memória
  - Baixo *bandwidth*
- **Para uso em dispositivos móveis e.g., PalmPilot's, telefones, etc.**

50