State of the Art
Application Development on GPUs

| Seoul, Korea | December 18, 2010 |
Phillip Miller, NVIDIA
Paul Arden, mental images
NVIDIA Resources for Application Developers

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS,CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS, CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS, CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS, CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS, CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

**DEVELOPMENT TOOLS**

- **CUDA Toolkit**
  Complete GPU computing development kit

- **cuda-gdb**
  GPU hardware debugging

- **Visual Profiler**
  GPU hardware profiler for CUDA C and OpenCL

- **Parallel Nsight**
  Integrated development environment for Visual Studio

- **NVPerfKit**
  OpenGL|D3D performance tools

- **FX Composer**
  Shader Authoring IDE

**VIDEO LIBRARIES**

- **Video Decode Acceleration**
  NVCUVID / NVCUVENC
  DXVA
  Win7 MFT

- **Video Encode Acceleration**
  NVCUVENC
  Win7 MFT

- **Post Processing**
  Noise reduction / De-interlace / Polyphase scaling / Color process

**ENGINES & LIBRARIES**

- **Math Libraries**
  CUFFT, CUBLAS, CUSPARSE, CURAND, ...

- **NPP Image Libraries**
  Performance primitives for imaging

- **App Acceleration Engines**
  Optimized software modules for GPU acceleration

- **Shader Library**
  Shader and post processing

- **Optimization Guides**
  Best Practices for GPU computing and Graphics development

**SDKs AND CODE SAMPLES**

- **GPU Computing SDK**
  CUDA C, OpenCL, DirectCompute code samples and documentation

- **Graphics SDK**
  DirectX & OpenGL code samples

- **PhysX SDK**
  Complete game physics solution

- **OpenAutomate**
  SDK for test automation

http://developer.nvidia.com
Licensed solutions from mental images

**Integrated Renderers**
- **mental ray**
  the world’s most widely adopted professional ray tracing solution
- **iray**
  The world’s first commercially available, physically correct rendering with GPU acceleration

**Material Workflows**
- **metaSL**
  Shading language extending from mental ray to real-time shader APIs
- **mental mill**
  Visual shader editor for end users to create and edit MetaSL shaders

**Application Building**
- **RealityServer**
  A 3D web services development platform supporting collaboration and a wealth of rendering options
- **neuray**
  Application foundation for building 3D applications with native couplings to mental images rendering solutions
- **mental matter**
  Higher order surface definition & approximation

**Distributed Processing**
- **DiCE**
  Highly scalable distributed processing solution for neuray applications

**More...**
Numerous renderers to fill particular needs.

www.mental.com
“GPGPU or GPU Computing”

- Using all processors in the system for the things they are best at doing:
  - Evolution of CPUs makes them good at sequential, *serial* tasks
  - Evolution of GPUs makes them good at *parallel* processing
CUDA - NVIDIA’s Architecture for GPU Computing

GPU Computing Applications

<table>
<thead>
<tr>
<th>CUDA C/C++</th>
<th>OpenCL</th>
<th>Direct Compute</th>
<th>Fortran</th>
<th>Python, Java, .NET, more…</th>
</tr>
</thead>
<tbody>
<tr>
<td>+100k developers</td>
<td>Commercial OpenCL Conformant Driver</td>
<td>Microsoft API for GPU Computing</td>
<td>PGI Accelerator</td>
<td></td>
</tr>
<tr>
<td>In production usage since 2008</td>
<td>Publicly Available for all CUDA capable GPU’s</td>
<td>Supports all CUDA-Architecture GPUs (DX10 and DX11)</td>
<td>PGI CUDA Fortran</td>
<td></td>
</tr>
<tr>
<td>SDK + Libs + Visual Profiler and Debugger</td>
<td>SDK + Visual Profiler</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NVIDIA GPU
with the CUDA Parallel Computing Architecture

Broad Adoption

- +250M CUDA-enabled GPUs in use
- +650k CUDA Toolkit downloads in last 2 Yrs
- +350 Universities teaching GPU Computing on the CUDA Architecture
- Cross Platform:
  Linux, Windows, MacOS
- Uses span
  HPC to Consumer

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.
Accelerating Existing Applications

1. Identify Possibilities
   - Profile for Bottlenecks,
     Inspect for Parallelism

2. Port Relevant Portion
   - A Debugger is a good starting point,
     Consider Libraries & Engines vs. Custom Code

3. Validate Gains
   - Benchmark vs. CPU version

4. Optimize
   - Parallel Nsight, Visual Profiler,
     GDB, Tau CUDA, etc.

5. Deploy
   - Maintain original as CPU fallback if desired.

Production Example
GPU Computing Software Stack

Your GPU Computing Application

Application Acceleration Engines
Middleware, Modules & Plug-ins

Foundation Libraries
Low-level Functional Libraries

Development Environment
Languages, Device APIs, Compilers, Debuggers, Profilers, etc.

CUDA Architecture
NVIDIA Application Acceleration Engines (AXE)

A family of highly optimized software modules, enabling software developers to supercharge applications with high performance capabilities that exploit NVIDIA GPUs.

- Free to acquire, license and deploy
- Valuable features and superior performance are quick to add
- App’s can evolve quickly, as API’s abstract GPU advancements
Application Acceleration Engines

**PhysX** *physics & dynamics engine*
- breathing life into real-time 3D; **Apex** enabling 3D animators

**Cg/CgFX** *programmable shading engine*
- enhancing realism across platforms and hardware

**SceniX*** scene management engine
- the basis of a real-time 3D system

**CompleX** scene scaling engine
- giving a broader/faster view on massive data

**OptiX** ray tracing engine
- making ray tracing ultra fast to execute and develop

*include bridges to external solutions - iray, MetaSL, OSG, OIV, etc.*
Accelerating Application Development

App Example: Auto Styling

1. Establish the Scene = SceniX

2. Maximize interactive quality + CgFX + OptiX

3. Maximize production quality + iray (licensed)

App Example: Seismic Interpretation

1. Establish the Scene = SceniX

2. Maximize data visualization + quad buffered stereo + volume rendering + ambient occlusion

3. Maximize scene size + CompleX
Acceleration Engine Relationships 2010

- AXE Connections
  - CgFX
    - Tessellation
  - QB Stereo
  - 30-bit color
  - GSync
  - SDI i/o

- Application Building
  - Quadro HiW Features
  - Scene Scaling Physics

- AXE Engines
  - OptiX
  - iray (licensed)
  - PhysX (coming)
  - CompleX

- AXE Reach
  - Non-Graphic Applications
  - Custom Scene Graphs & Real-time
  - Open Scene Graph
  - VSG’s Open Inventor

- SceniX
  - Ray Tracing, GL
  - CgFX
  - OpenGL

- VSG’s Open Scene Graphs & Real-time
SceniX™ Scene Management Engine

The fastest start for building a real-time 3D app - wherever there’s a need to analyze 3D data, make decisions, and convey results in real-time

- Highly efficient scene graph for rapidly building real-time 3D app’s for any OpenGL GPU on Windows/Linux
- Integration interface for using GUI frameworks (Qt, wxWidgets, etc.)
- Fast on-ramp to GPU capabilities & NVIDIA engines
  - Quad Buffered Stereo, SDI i/o, 30-bit color, etc.
  - CgFX, CompleX, OptiX, Tessellation
- Source Code license available (upon approval)
- Differentiator - Multiple Render Targets

Showcase images courtesy Autodesk
SceniX - Example Companies/Products
+5k downloads/version
v6 in July

RTT
challenging reality

Autodesk

DeltaGen
Showcase

LIGHTWORKS
Rendering Realism

AUTHOR\textsuperscript{LW}
ASPECTS\textsuperscript{LW}
ARTISAN\textsuperscript{LW}
SceniX and CgFX example

- Quadro 6000 Demo Viewer
SceniX - Renderer Independency

- Separates rendering from destination (tiles, cameras, viewports, renderers, image gen, etc.)
- Multiple render engines within a single render window
- Together enabling:
  - Stack Rendering (multiple techniques and renderers)
  - Hybrid Rending (raster + ray tracing)
  - Post Processing
  - Platform Impendence
Stack Rendering Example

- Combining two different renderers to create realistic reflections on top of an OpenGL rendered object
Hybrid Rendering - more results
Multiple Rendering Example

- New Demo Viewer coming in 2011 with Multiple Rendering Capabilities
- Coordinates shader usage between OpenGL, CgFX, OptiX and iray
- Cross platform, using Qt
- Source will be available to registered developers
CompleX™ Scene Scaling Engine

Keeps complex scenes interactive as they exceed single GPU memory, by managing the combined memory and performance of multiple GPUs

Delivers smooth performance on very large scenes:
- 32GB in size on Quadro FX 5800
- 48GB in size on Quadro 6000

SDK for any OGL application
- Ready to use in SceniX, OpenSceneGraph, and Open Inventor 8.1
SceniX and CgFX example

- Quadro 6000 Demo Viewer
CompleX™ Example Companies

National Institute of Health

VSG Open Inventor

StormFjord & Statoil
CompleX™ - Distribute & Composite

Made of two components, that can be used independently:

- Data Distribution
  - slicing scenes across GPUs to keep them within frame buffer memory
- Image Compositing
  - the fastest available image combination from multiple GPU outputs
- Multiple approaches for each component to accommodate different data and transparency needs
CompleX™ - Methods

>500 million pixels/second

Screen Compositing

Depth Compositing

Alpha Compositing
CompleX™ - Composite

The industry’s fastest multi-GPU compositor (no SLI req’d)

- Uses unique NVIDIA hw/driver features `copy_tex_image` across multiple GPUs
- Highly optimized for GPU to GPU: multiple transfer paths optimized for a wide variety of multi-GPU and chipset configurations.
- Results in the best performance for given HW
- Resulting event loop typically needs +2 lines of code
NVIDIA OptiX™ Ray Tracing Engine

A programmable ray tracing pipeline for accelerating interactive ray tracing applications - from functions, to tasks, to complete renderers. In use within a wide variety of markets - not just rendering

- For Windows, Linux, and OSX on all CUDA capable GPUs
- C-based shaders/functions (minimal CUDA exp. needed)
- Ease of Development - you concentrate on writing ray tracing techniques, and OptiX makes them fast

Applications benefit immediately from GPU advances:
- Highly scalable on cores and GPUs - SLI not required
- GPU advances - GF100 is 2-4X of GT200 which is 2X of G80
- OptiX advances - 2.1 (this week) +30 to 80% faster than 2.0
OptiX™ - SDK Examples

- Whitted
- Cook
- Photon Mapping
- Glass
- Fish Tank
- Collision Detection
- Modified SDK Example - MandleBulb
- Fast AO
OptiX™ - Example Customers
+3k downloads / version

Privately at major companies doing:
• Radiation & Magnetic Reflection
• Acoustics and Ballistics
• Multi-Spectral Simulation
• Motion Picture production
• Massive On-Line Player Games
NVIDIA Design Garage Demo

- Photorealistic car configurator in the hands of millions of consumers

- Uses pure GPU ray tracing
  - 3-4X faster on GF100 than on GT200
  - Linear scaling over GPUs & CUDA Cores
  - Est. 40-50X faster vs. a CPU core

- Built on SceniX with OptiX shaders
  - similar to other apps in development

- Rendering development speed
  - 6 weeks
Application Engine Availability

nvidia.com

Developer Zone
iray® from mental images

World’s first commercial, physically correct, interactive global illumination renderer - greatly speeding the creative workflow for designers with intuitive results that match the real world.

Scalable across processors and nodes for maximum interactivity. Many times faster on GPUs than CPU.

Availability:
- w/ mental ray® 3.8 & RealityServer
- stand-alone Integrator Edition
- Coming to SceniX in 2011
- Integrated in Bunkspeed Shot, Autodesk 3ds Max 2011, DS Catia v6
iray is the new CUDA-accelerated rendering mode inside mental ray 3.8, RealityServer 3.0 and other products.

- See full global illumination effects in seconds
- Quickly preview final frame quality in selected image areas
- Work without learning render-specific parameters
- Render final frames with complex global illumination effects much faster than CPU renderers
- Less overhead from tuning scenes and shaders
The server based architecture of RealityServer give the following key advantages over traditional client-side technologies:

- Independence from Data Complexity
- Thin Clients
- Collaboration
- Data security
- Scalability
- Development Choice
- State of the art Rendering
GPU Cloud Computing

A significant trend is arising towards Cloud Computing for large scale deployments. RealityServer is ideal for Cloud Computing:

- Successfully deployed on:
  - Amazon EC2
  - PEER 1
  - Penguin Computing

- Web Services significantly ease communication with other Cloud resources or off-Cloud resources

- Straightforward way to scale with RealityServer resource requirements
RealityServer is built on our proprietary DiCE technology. It is ideally suited to Cloud based deployments:

- Master-less self-organizing cluster architecture
- Fault-tolerant in-memory distributed database
- Automated load balancing across resources (CPUs, GPUs)
- Dynamically add and remove computing resources
- Large scale clustering over GbE and 10GbE networks
- Multi-user by design
- Targeting very low latencies and large numbers of jobs
- Cloud specific clustering modes for Unicast only networks
Thank you!

- Questions?