Languages, APIs and Development Tools for GPU Computing

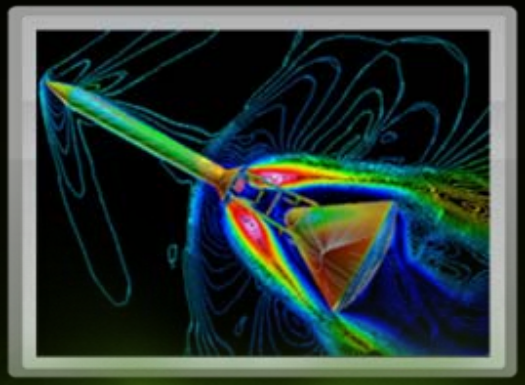
Phillip Miller
Director, Software Product Management
Professional Solutions Group

NVIDIA

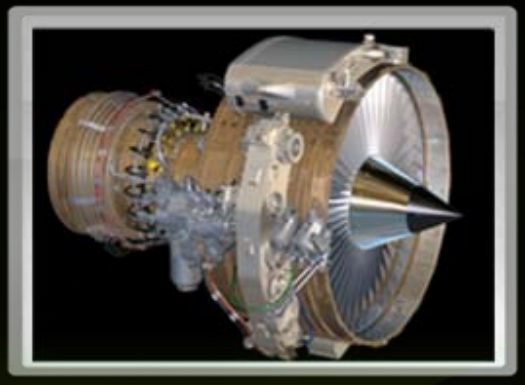
CUDA Architecture
Available on All Modern NVIDIA GPUs

GeForce®
Entertainment

Tesla™
High Performance Computing



Quadro®
Design & Creation

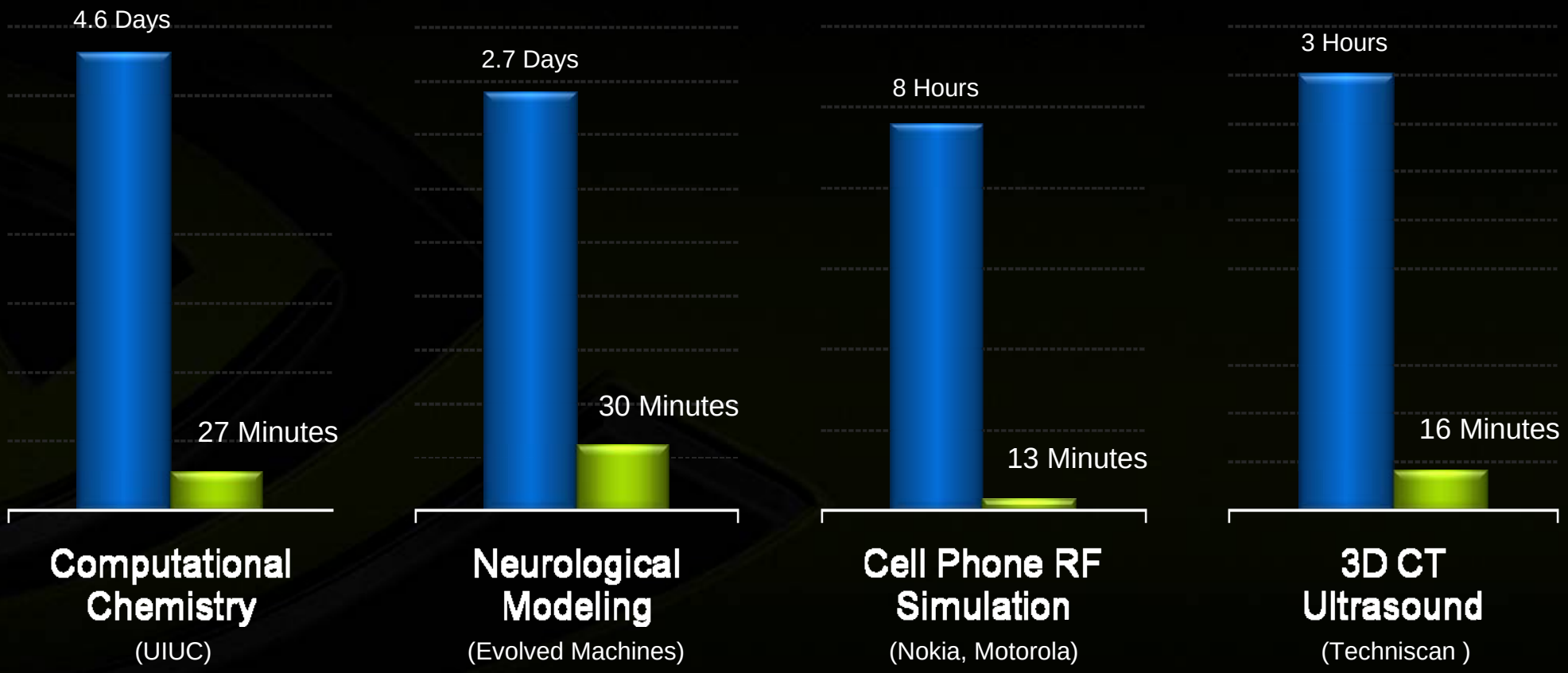


Over 150 Million CUDA-enabled GPUs Installed

© 2009 NVIDIA Corporation

NVIDIA.

GPUs are Accelerating Time to Discovery



NVIDIA - CONFIDENTIAL

© 2009 NVIDIA Corporation

Huge Speed-Ups Across Many Fields

Algorithm	Field	Speedup
2-Electron Repulsion Integral	Quantum Chemistry	**130X**
Lattice Boltzmann	CFD	**123X**
Euler Solver	CFD	**16X**
GROMACS	Molecular Dynamics	**137X**
Lattice QCD	Physics	**30X**
Multifrontal Solver	FEA	**20X**
nbody	Astrophysics	**100X**
Simultaneous Iterative Reconstruction Technique	Computed Tomography	**32X**

© 2009 NVIDIA Corporation

NVIDIA.

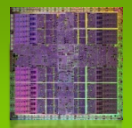
GPU Computing Overview

Broad Adoption

- Over 150,000,000 installed CUDA-Architecture GPUs
- Over 90,000 GPU Computing Developers (9/09)
- Windows, Linux and MacOS Platforms supported
- GPU Computing spans HPC to Consumer
- 250+ Universities teaching GPU Computing on the CUDA Architecture

GPU Computing Applications

CUDA C/C++	OpenCL	Direct Compute	Fortran	Python, Java, .NET, …
• Over 90,000 developers • Running in Production since 2008 • SDK + Libs + Visual Profiler and Debugger	• 1st GPU demo • Shipped 1st OpenCL Conformant Driver • Public Availability (Since April)	• Microsoft API for GPU Computing • Supports all CUDA-Architecture GPUs (DX10 and DX11)	• PGI Accelerator • PGI CUDA Fortran • NOAA Fortran bindings • FLAGON	• PyCUDA • jCUDA • CUDA.NET • OpenCL.NET



NVIDIA GPU
with the CUDA Parallel Computing Architecture

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

© 2009 NVIDIA Corporation

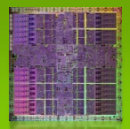
GPU Computing Application Development

Your GPU Computing Application

Application Acceleration Engines (AXEs)
Middleware, Modules & Plug-ins

Foundation Libraries
Low-level Functional Libraries

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

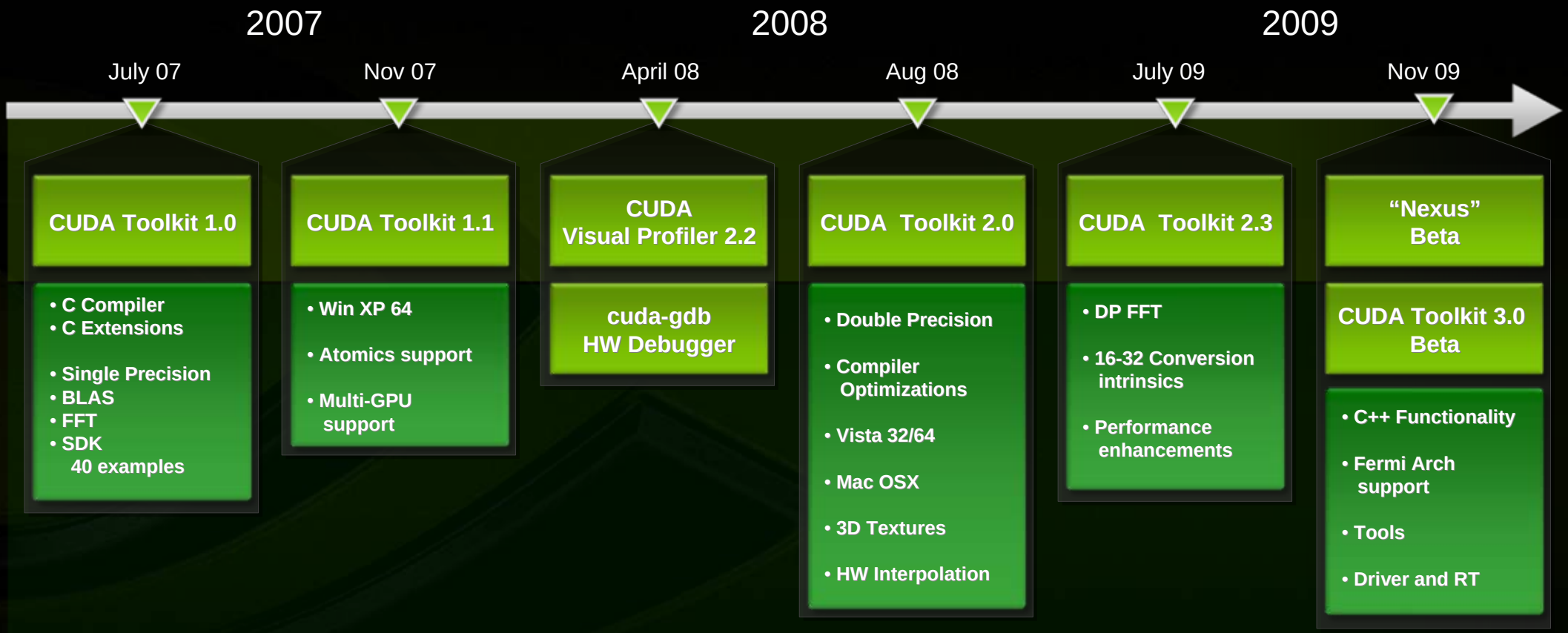


CUDA Architecture

© 2009 NVIDIA Corporation


NVIDIA.
Languages & APIs

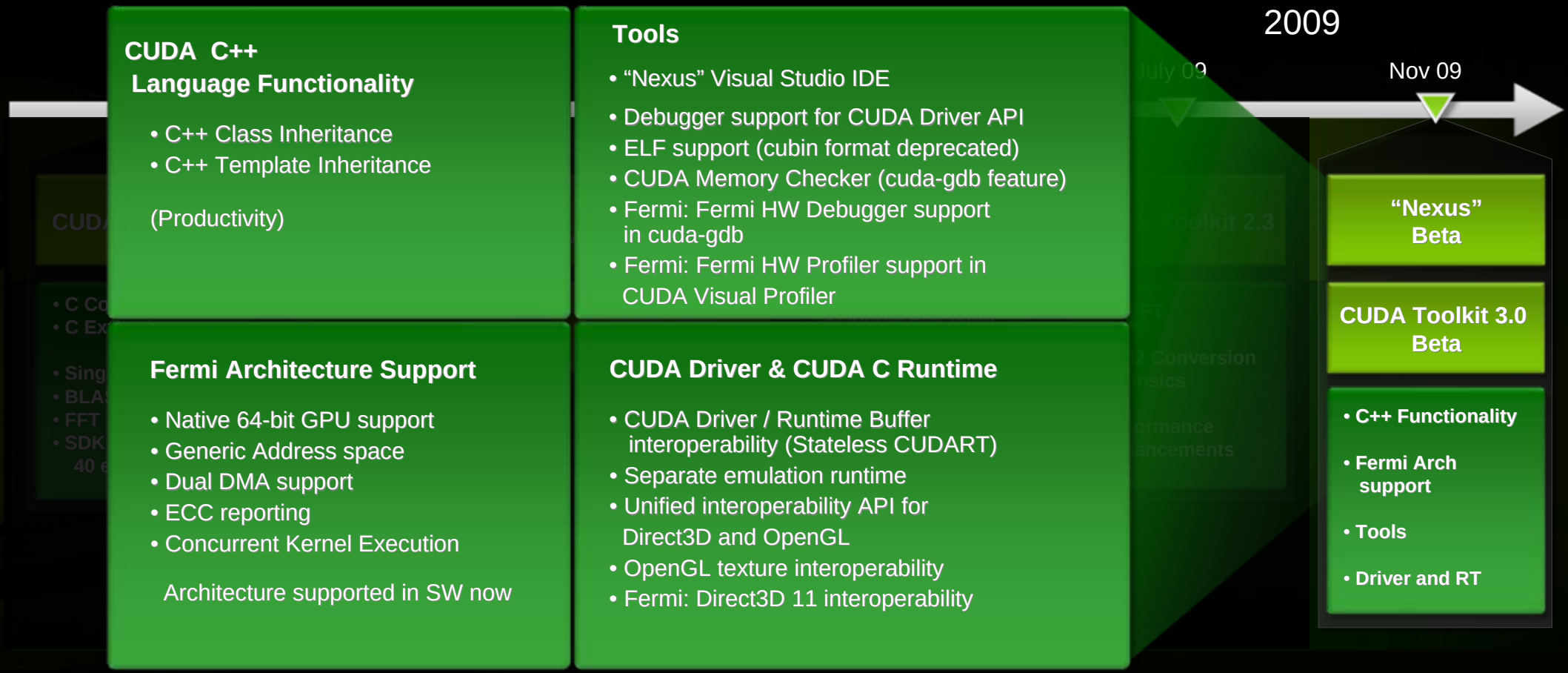
CUDA C/C++ Update



© 2009 NVIDIA Corporation

NVIDIA.

CUDA C/C++ Update



© 2009 NVIDIA Corporation

NVIDIA

Fortran Language Solutions

- 3 PGI Accelerators
  - High-level *implicit* programming model, similar to OpenMP
  - Auto-parallelizing compiler
- 4 PGI CUDA Fortran Compiler
  - High-level *explicit* programming model, similar to CUDA C Runtime
- 1 NOAA F2C-ACC
  - Converts Fortran codes to CUDA C
  - Some hand-optimization expected
- 2 FLAGON
  - Fortran 95 Library for GPU Numerics
  - Includes support for cuBLAS, cuFFT, CUDPP, etc.

© 2009 NVIDIA Corporation

OpenCL

- Cross-vendor open standard
  - Managed by the Khronos Group
- Low-level API for device management and launching kernels
  - Close-to-the-metal programming interface
  - JIT compilation of kernel programs
- C-based language for compute kernels
  - Kernels must be optimized for each processor architecture

http://www.khronos.org/opencl

NVIDIA released the first OpenCL v1.0 conformant driver for Windows and Linux to thousands of developers in June 2009

© 2009 NVIDIA Corporation

NVIDIA OpenCL Support

R195

OpenCL ICD

OpenGL Interoperability

Double Precision

R190

NVIDIA Compiler Optimization Flags

Query for Compute Capability

Byte Addressable Stores

32-bit Atomics

Images

Conformance

OpenCL 1.0 Driver
Min. Spec. NV R190 released June 2009

NVIDIA OpenCL Visual Profiler

NVIDIA OpenCL Documentation

NVIDIA OpenCL Code Samples

© 2009 NVIDIA Corporation

NVIDIA

DirectCompute

- Microsoft standard for all GPU vendors
  - Released with DirectX® 11 / Windows 7
  - Runs on all 150M+ CUDA-enabled DirectX 10 class GPUs and later
- Low-level API for device management and launching kernels
  - Good integration with other DirectX APIs
- Defines HLSL-based language for compute shaders
  - Kernels must be optimized for each processor architecture

© 2009 NVIDIA Corporation

Language & APIs for GPU Computing

Solution	Approach
CUDA C / C++	Language Integration Device-Level API
PGI Accelerator PGI CUDA Fortran	Auto Parallelizing Compiler Language Integration
CAPS HMPP	Auto Parallelizing Compiler
OpenCL	Device-Level API
DirectCompute	Device-Level API
PyCUDA	API Bindings
CUDA.NET, OpenCL.NET, jCUDA	API Bindings
…	…

© 2009 NVIDIA Corporation

NVIDIA.


NVIDIA.
Development Tools

NVIDIA Developer Resources

DEVELOPMENT TOOLS

CUDA Toolkit
Complete GPU computing development kit

Visual Profiler
GPU hardware profiler for CUDA C and OpenCL

Nexus
Development environment with Visual Studio integration [*beta*]

NVPerfKit
OpenGL|D3D performance tools

FX Composer
Shader Authoring IDE

SDKs AND CODE SAMPLES

GPU Computing SDK
CUDA C, OpenCL, DirectCompute programming guide and code samples

Graphics SDK
DirectX & OpenGL code samples

PhysX SDK
Complete game physics solution

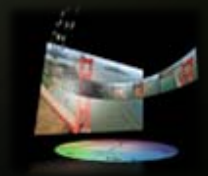
OpenAutomate
Test automation SDK

VIDEO LIBRARIES

Video Decode Acceleration
NVCUVID
DXVA
Win7 MFT

Video Encode Acceleration
NVCUVENC
Win7 MFT

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



ENGINES & LIBRARIES

NPP Image Libraries
Performance primitives for imaging

Numeric Libraries
cuFFT, cuLA, cuBLAS

App Acceleration Engines
Optimized software modules for GPU acceleration

Shader Library
Shader and post processing

Optimization Guides
GPU computing and Graphics development best practices

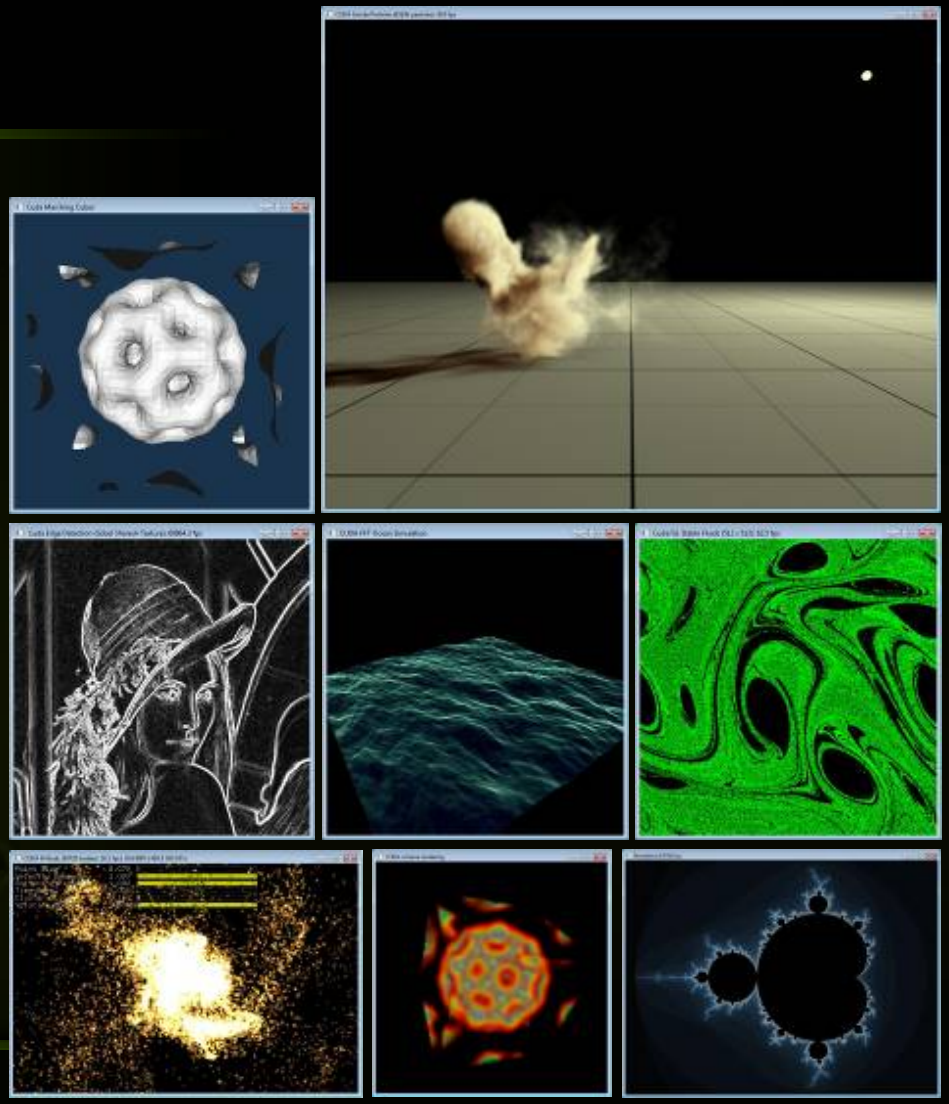
© 2009 NVIDIA Corporation

NVIDIA.

NVIDIA SDKs

Hundreds of code samples for CUDA C/C++, DirectCompute, and OpenCL

- Finance
- Oil & Gas
- Video/Image Processing
- 3D Volume Rendering
- Particle Simulations
- Fluid Simulations
- Math Functions



© 2009 NVIDIA Corporation

NVIDIA.

NVIDIA Application Acceleration Engines

OptiX – ray tracing engine

- Programmable GPU ray tracing pipeline that greatly accelerates general ray tracing tasks
- Supports programmable surfaces and custom ray data

PhysX – physics and dynamics engine

- GPU computed physics – fast enough for real time.
- In use across games, DCC tools, and simulation

SceniX– scene management engine

- High performance OpenGL scene graph built around CgFX for maximum interactive quality
- Provides ready access to new GPU capabilities & engines

OptiX shader example

PhysX particles example

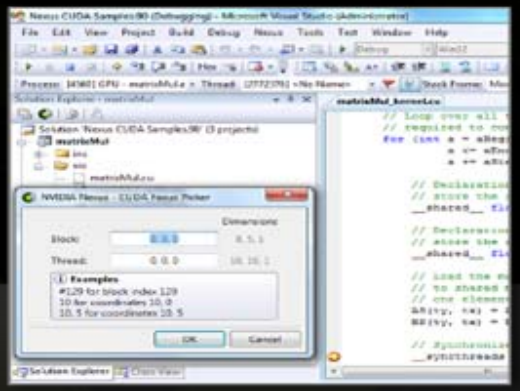
Autodesk Showcase customer example

© 2009 NVIDIA Corporation

NVIDIA.

“Nexus” 1.0 Beta

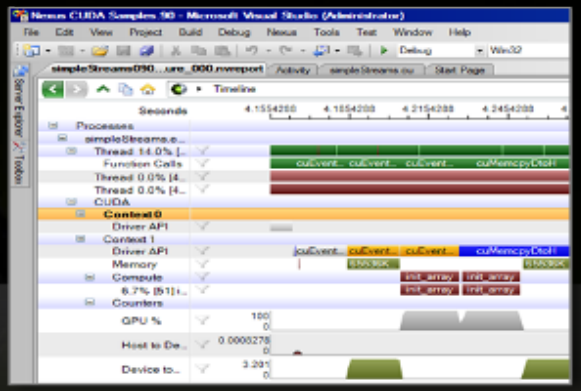
Microsoft Visual Studio PARTNER



Parallel Debugger

GPU source code debugging

Variable & memory inspection

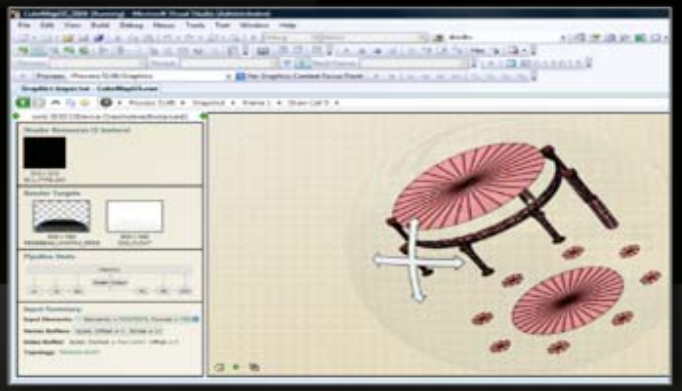


System Analyzer

Platform-level Analysis

For the CPU and GPU

Visualize Compute Kernels, Driver API Calls, and Memory Transfers



Graphics Inspector

Visualize and debug graphics content

© 2009 NVIDIA Corporation

NVIDIA.

Massively Parallel Development

Tools for Windows

	Visual Studio Integration	Parallel Debugging		Parallel Profiling		System Analysis/ Trace (CPU/GPU)		Premium Support	Early Access	Multi-Vendor GPU Support
		Compute	Gfx	Compute	Gfx	Compute	Gfx			
“Nexus”* Pro	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
“Nexus”*	✓	✓	✓	✓	✓					
NV Visual Profiler				✓						
Mcore Platform Analyzer				✓		✓				
PerfKit/PerfHUD			✓		✓		✓			
gDEBugger			✓		✓		✓	✓		✓

* “Nexus” is NVIDIA’s code name

© 2009 NVIDIA Corporation

NVIDIA.

Massively Parallel Development

Tools for Linux

	Compilation	Debugging	Profiling	Analysis	Premium Support
CAPS HMPP	✓				✓
PGI Accelerators	✓				✓
PGI CUDA Fortran	✓				✓
NV cuda-gdb		✓			
Allinea DDT		✓			✓
TotalView Debugger		✓			✓
NV Visual Profiler			✓		
TAU CUDA			✓	✓	
Mcore Platform Analyzer			✓	✓	✓

cuda-gdb will be extended to support OpenCL debugging in a future release, with solutions from Allinea, TotalView and others expected to follow.

© 2009 NVIDIA Corporation

NVIDIA.

Massively Parallel Development

Tools for Linux

	Debugging	Profiling	Analysis	Cluster Support	Lib's
CUDA C/C++	✓	✓	✓	✓	✓
OpenCL	Coming Soon	✓	Coming Soon	✓	Coming Soon
Fortran		✓	✓	✓	✓

cuda-gdb will be extended to support OpenCL debugging in a future release, with solutions from Allinea, TotalView and others expected to follow.

© 2009 NVIDIA Corporation

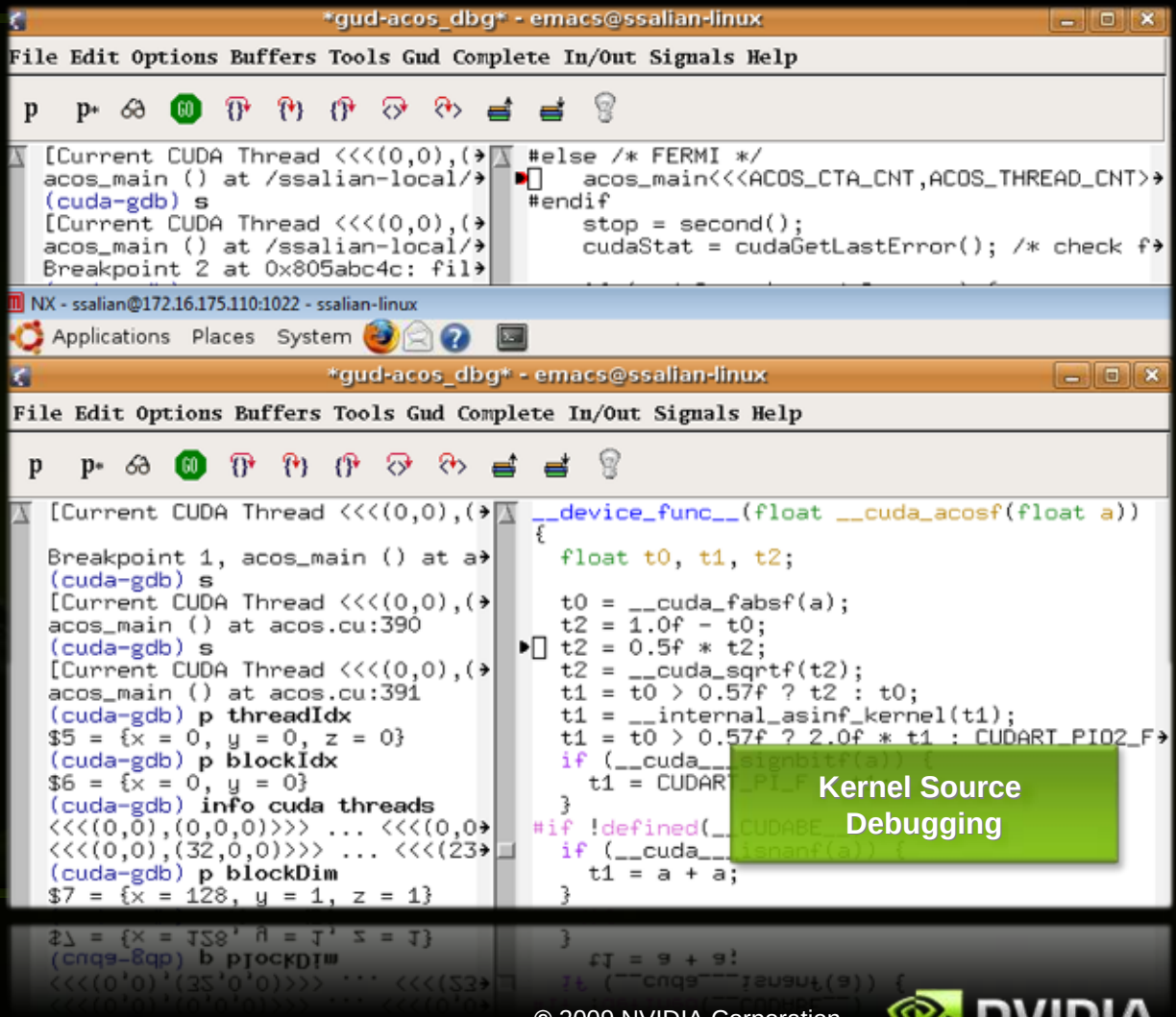
NVIDIA.

cuda-gdb

CUDA debugging integrated into GDB on Linux

- Supported on 32bit and 64bit systems
- Seamlessly debug both the host/CPU and device/GPU code
- Set breakpoints on any source line or symbol name
- Access and print all CUDA memory allocs, local, global, constant and shared vars

Included in the CUDA Toolkit



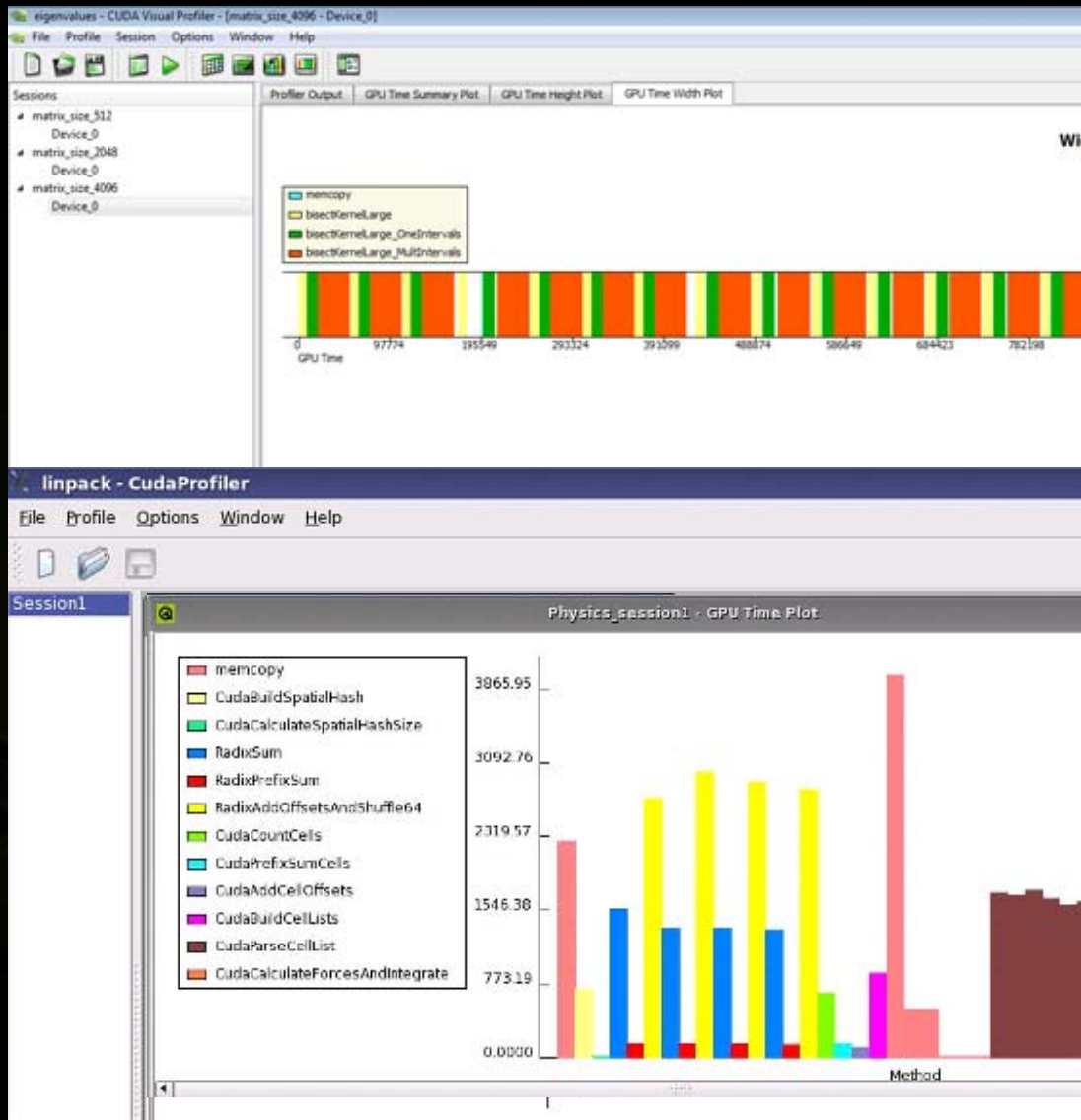
© 2009 NVIDIA Corporation

NVIDIA.

CUDA Visual Profiler

- **Analyze GPU HW performance** signals, kernel occupancy, instruction throughput, and more
- **Highly configurable** tables and graphical views
- **Save/load profiler sessions or export to CSV for later analysis**
- **Compare results visually** across multiple sessions to see improvements
- **Windows, Linux and Mac OS X supported**
  OpenCL Visual Profiler for Windows and Linux

Included in the CUDA Toolkit



Massively Parallel Development (summary)

Tools for Linux - Language / API Support

	CUDA C/C++	OpenCL	Fortran
CAPS HMPP	✓		✓
PGI Accelerators	✓		✓
PGI CUDA Fortran	✓		✓
NV cuda-gdb	✓		
Allinea DDT	✓		
TotalView Debugger	✓		
NV Visual Profiler	✓	✓	
TAU CUDA	✓		✓
Mcore Platform Analyzer	✓		✓

cuda-gdb will be extended to support OpenCL debugging in a future release, with solutions from Allinea, TotalView and others expected to follow.

© 2009 NVIDIA Corporation

NVIDIA.

200+ Universities Teaching GPU Computing on CUDA

© 2009 NVIDIA Corporation

NVIDIA.


Thank You
NVIDIA