Languages, APIs and Development Tools for GPU Computing

Will Ramey
Product Manager, GPU Computing

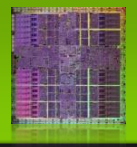
GPU Computing Overview

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	• 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

NVIDIA

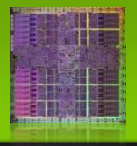
GPU Computing Application Development

Your GPU Computing Application

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

Foundation Libraries
Low-level Functional Libraries

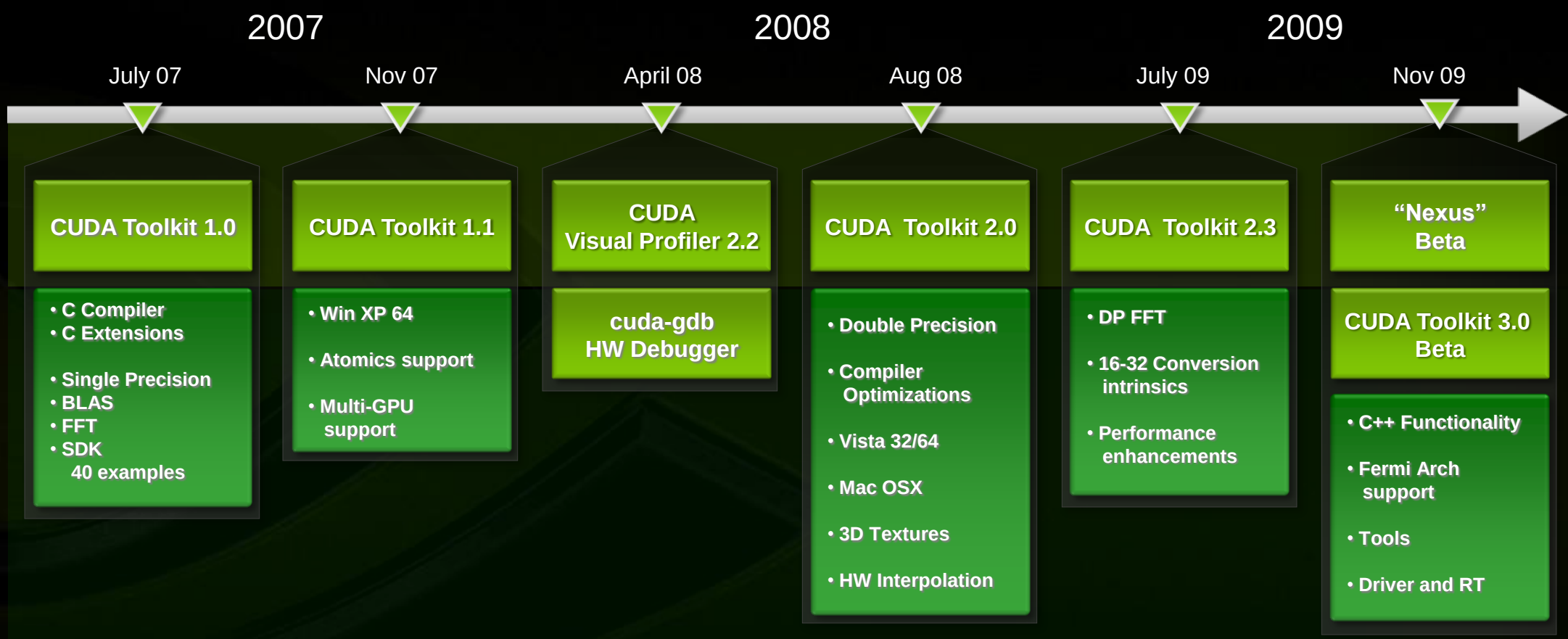
Development Environment
Languages, Device APIs, Compilers, Debuggers, Profilers, etc.



CUDA Architecture

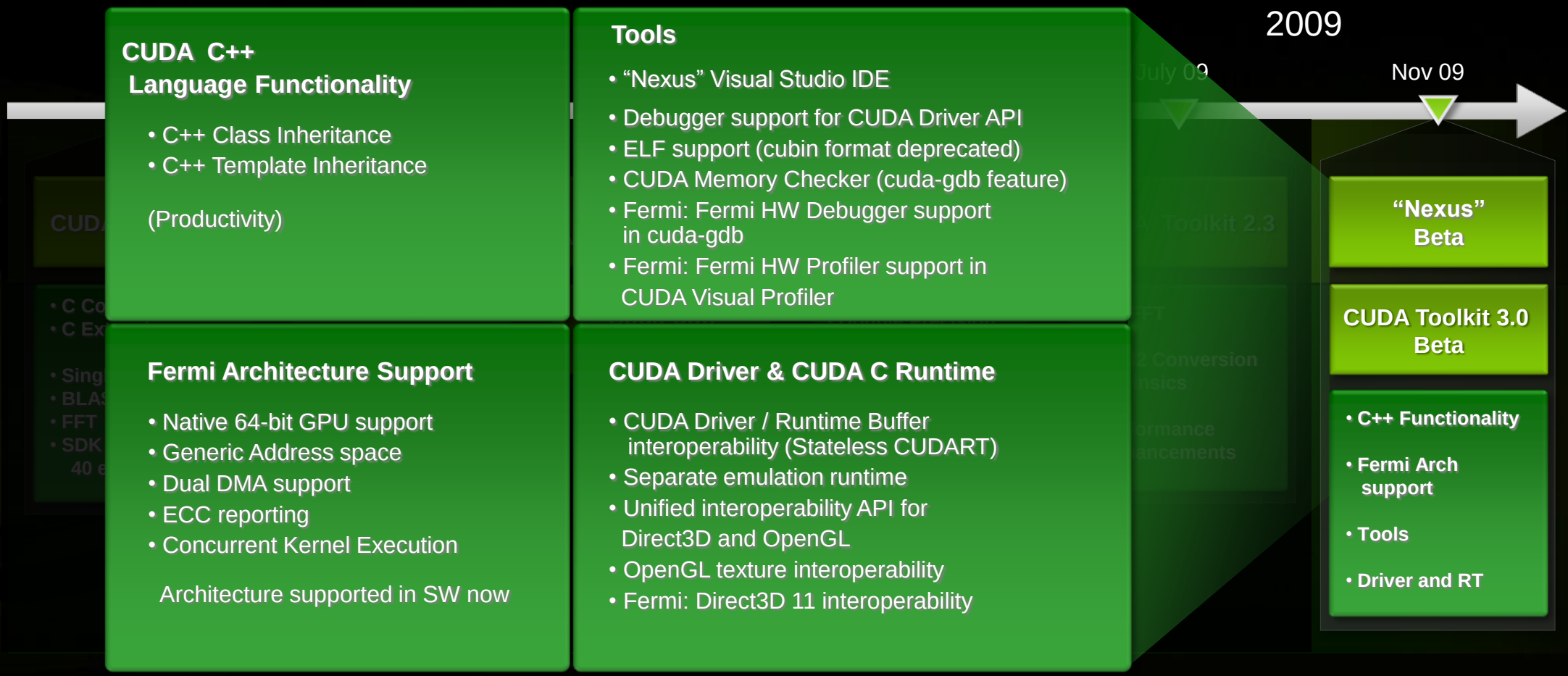

NVIDIA
Languages & APIs

CUDA C/C++ Update



NVIDIA

CUDA C/C++ Update



Fortran Language Solutions

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

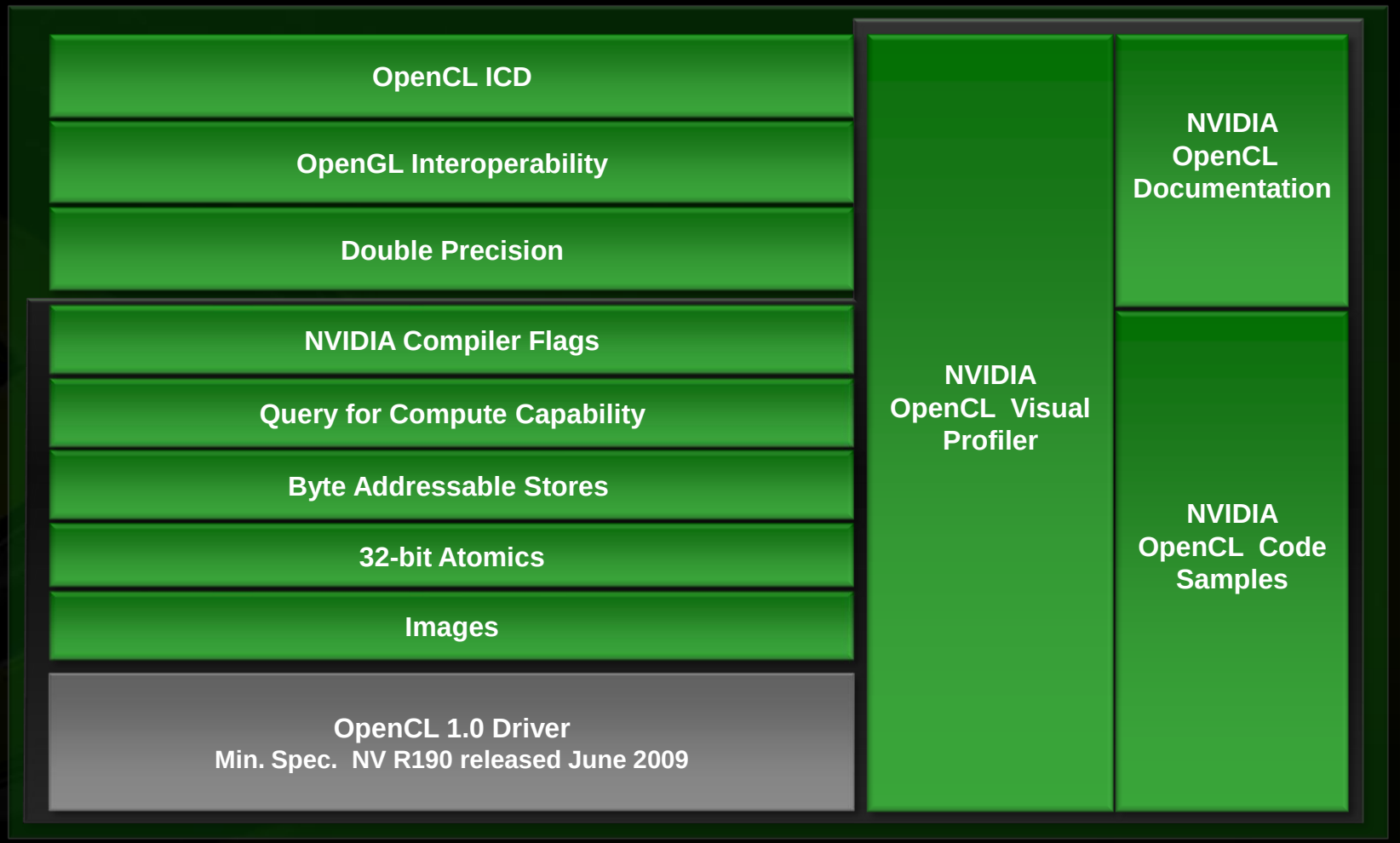
OpenCL

- 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

NVIDIA OpenCL Support



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

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
…	…

Massively Parallel Development

Language / API Support (Linux)

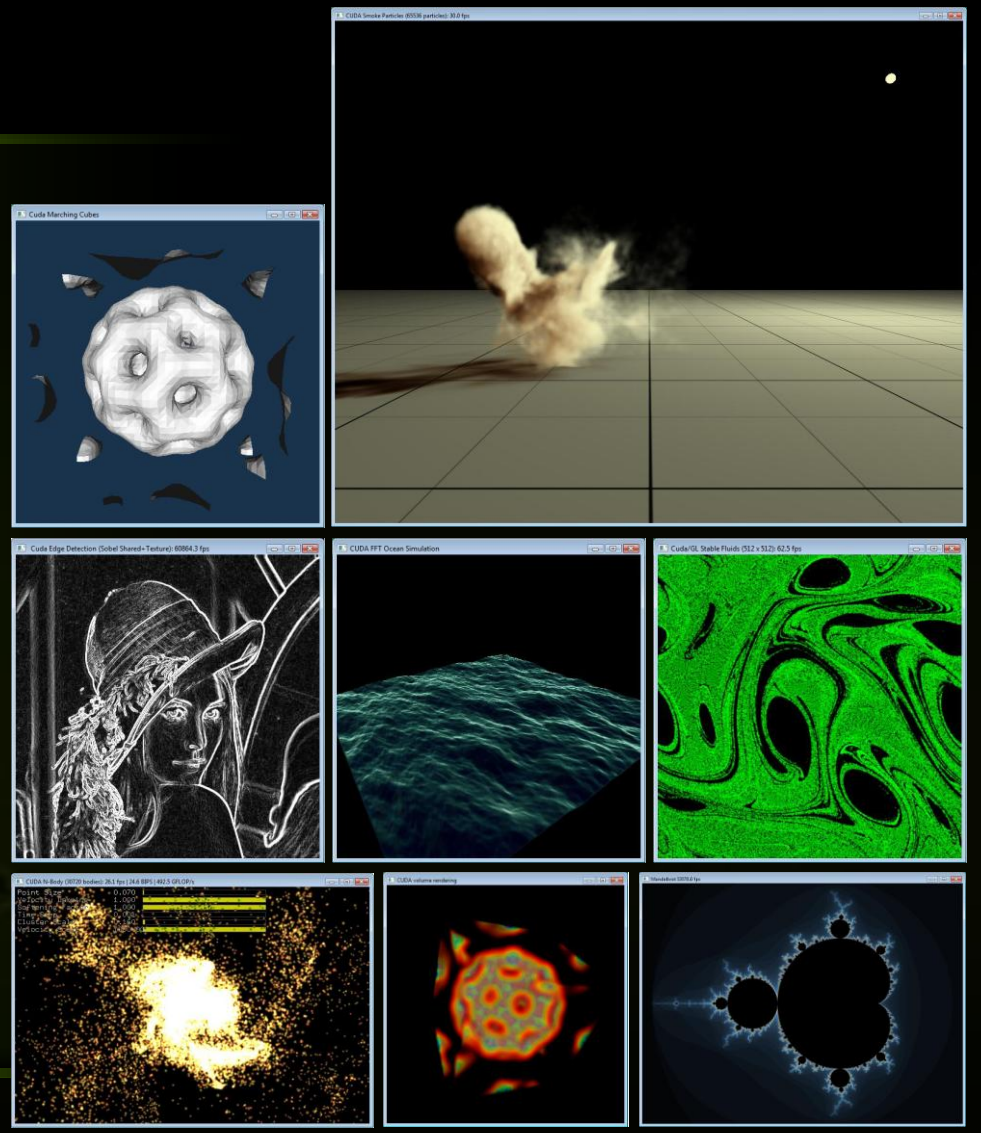
	CUDA C/C++	OpenCL	Fortran
CAPS HMPP	✓		✓
PGI Accelerators	✓		✓
PGI CUDA Fortran	✓		✓
NV cuda-gdb	✓		
Allinea DDT	✓		
TotalView Debugger	✓		
NV Visual Profiler	✓	✓	
TAU CUDA	✓		✓
Mcore Platform Analyzer	✓		✓

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

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



NVIDIA


NVIDIA
Development Tools

Massively Parallel Development

Tools for Linux

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

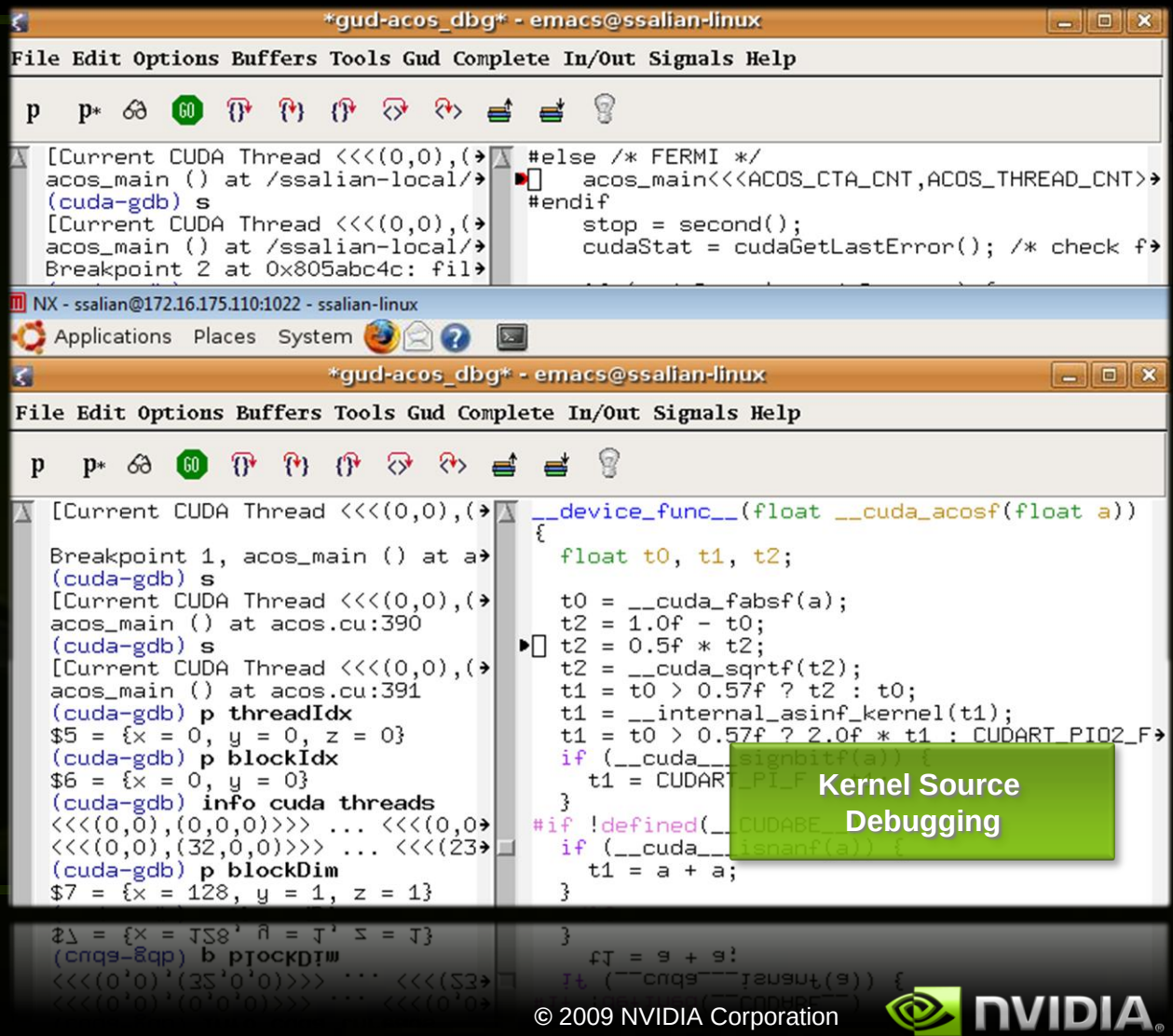
cuda-gdb will be extended to support OpenCL debugging in a future release, with solutions from Allinea, TotalView and others expected to follow.

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

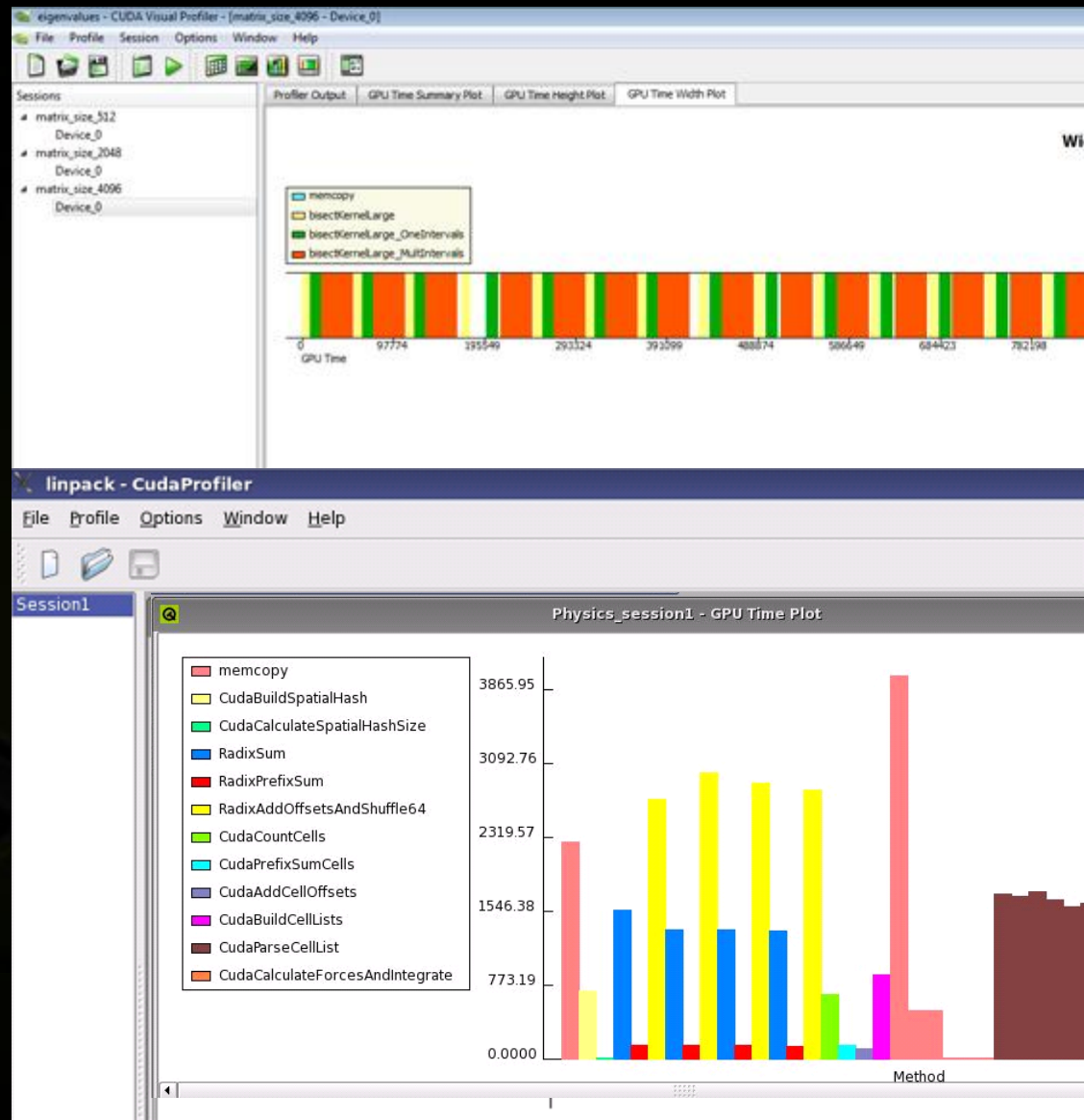
Included in the CUDA Toolkit



CUDA Visual Profiler

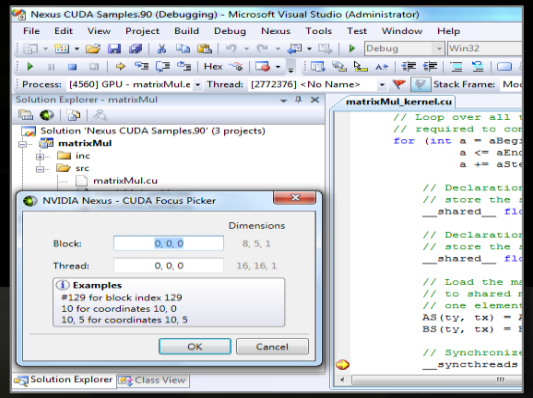
- **Analyze GPU HW performance** signals, kernel occupancy, instruction throughput, and more
- **Highly configurable** tables and graphical views
- **Save/load profiler sessions or export to CSV for later analysis**
- **Compare results visually** across multiple sessions to see improvements
- **Windows,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

Included in the CUDA Toolkit



“Nexus” 1.0 Beta

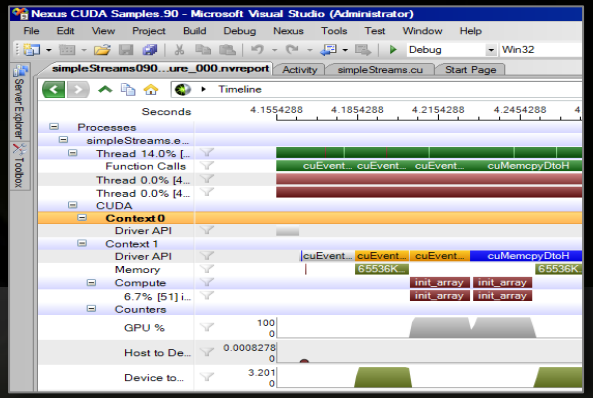
Microsoft Visual Studio PARTNER



Parallel Debugger

GPU source code debugging

Variable & memory inspection

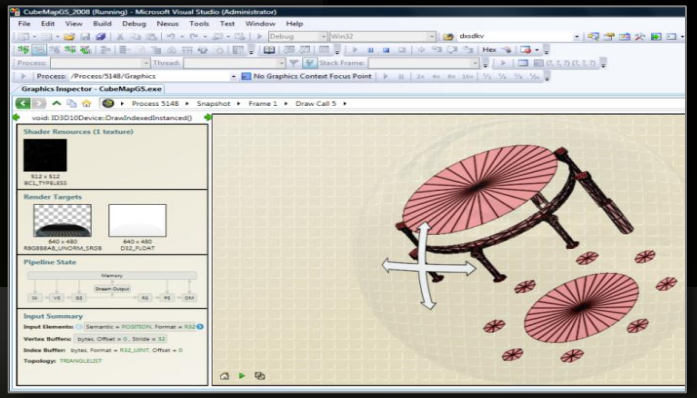


System Analyzer

Platform-level Analysis

For the CPU and GPU

Visualize Compute Kernels, Driver API Calls, and Memory Transfers



Graphics Inspector

Visualize and debug graphics content

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

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.

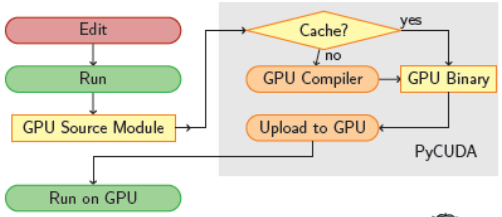

Thank You
NVIDIA

PyCUDA

Python + CUDA = PyCUDA

- All of CUDA in a modern scripting language
- Full Documentation
- Free, open source (MIT)
- Also: PyOpenCL

- CUDA C Code = Strings
- Generate Code Easily
  - Automated Tuning
- Batteries included: GPU Arrays, RNG, ...
- Integration: `numpy` arrays, Plotting, Optimization, ...



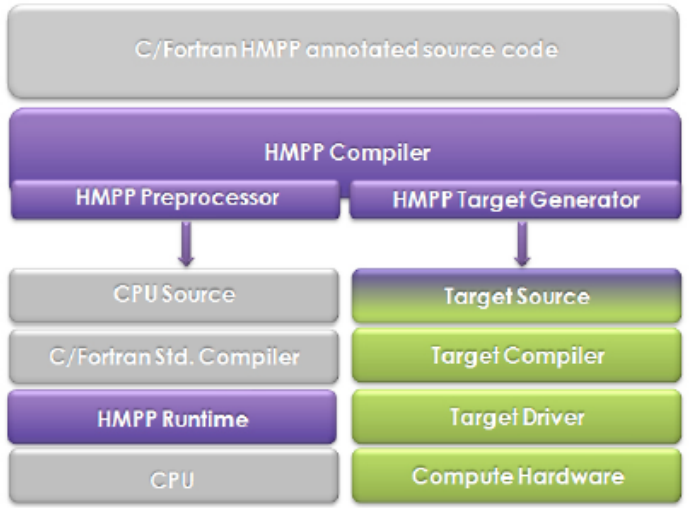
BROWN

Slide courtesy of Andreas Klöckner, Brown University

NVIDIA

HMPP Workbench Overview

- C and Fortran GPU programming directives
  - Define and execute GPU-accelerated versions of functions
  - Implement efficient communication patterns
  - Build parallel hybrid applications with OpenMP and MPI
- An open hybrid compiling workbench
  - Automatically generate CUDA computations
  - Use standard compilers and hardware vendor tools
  - Drive the whole compilation workflow
- A runtime library
  - Dispatch computations on available GPUs
  - Scale to multi-GPUs systems



CAPS

http://www.caps-entreprise.com/hmpp.html

NVIDIA

jCUDA

Using jCUDA you can create cross-platform CUDA solutions, that can run on any operating system supported by CUDA without changing your code.
Either select between Windows XP or Vista by Microsoft or even Linux/MacOS/Solaris systems.
Current support is for both 32 and 64 bits of every platform.

Features

- Double precision
- Object model for CUDA programming
- CUDA 2.1 Driver API
- CUDA 2.1 Runtime API
- CUFFT routines
- OpenGL interoperability
- * Support for CUBLAS routines will be added in the future

Operating System Support

- Microsoft Windows
- Linux
- * Support for Mac OSX will be added in the future
- * Support for Solaris 10 (x86) will be added in the future

http://www.hoopoe-cloud.com/Solutions/jCUDA/Default.aspx

Courtesy of Company for Advanced Supercomputing Solutions, Ltd.

CUDA.NET

2.3.6, 14/9/2009

Updates to native wrappers, added *SizeT* structure to handle 32/64 systems compatibility for functions taking *size_t* as parameter.
Support for CUDA 2.3 through .NET bindings to CUDA functions.
Currently supported on Windows, Linux and MacOS platforms.

Features

- Double precision
- Object model for CUDA programming
- CUDA 2.2 Driver API
- CUDA 2.2 Runtime API
- CUFFT routines
- CUBLAS routines
- Direct3D interoperability
- OpenGL interoperability

Operating System Support

- Microsoft Windows
- Linux 32/64 bit (using Mono)
- Mac OSX (using Mono)

http://www.hoopoe-cloud.com/Solutions/CUDA.NET/Default.aspx

Courtesy of Company for Advanced Supercomputing Solutions, Ltd.

OPENCL.NET

2.3.6, 14/9/2009
Updates to native wrappers, added *SizeT* structure to handle 32/64 systems compatibility for functions taking *size_t* as parameter.
Support for CUDA 2.3 through .NET bindings to CUDA functions.
Currently supported on Windows, Linux and MacOS platforms.

Features

- Double precision
- Object model for CUDA programming
- CUDA 2.2 Driver API
- CUDA 2.2 Runtime API
- CUFFT routines
- CUBLAS routines
- Direct3D interoperability
- OpenGL interoperability

Operating System Support

- Microsoft Windows
- Linux 32/64 bit (using Mono)
- Mac OSX (using Mono)

http://www.hoopoe-cloud.com/Solutions/CUDA.NET/Default.aspx

Courtesy of Company for Advanced Supercomputing Solutions, Ltd.