What’s New in CUDA 5

Mark Ebersole
Computer Vision
The Soul of CUDA

The Platform for High Performance Parallel Computing

Accessible High Performance
Enable Computing Ecosystem
GPU Accelerated Libraries
“Drop-in” Acceleration for your Applications

- NVIDIA cuBLAS
- NVIDIA cuRAND
- NVIDIA cuSPARSE
- NVIDIA NPP
- GPU VSIPL
- CULA tools
- MAGMA
- NVIDIA cuFFT
- Rogue Wave Software
- ArrayFire
- CUSP
- Thrust

Vector Signal Image Processing
GPU Accelerated Linear Algebra
Matrix Algebra on GPU and Multicore
IMSL Library
ArrayFire Matrix Computations
Sparse Linear Algebra
C++ STL Features for CUDA

© 2012 NVIDIA
Introducing
CUDA 5
CUDA 5
Application Acceleration Made Easier

Dynamic Parallelism
Spawn new parallel work from within GPU code on GK110

GPU Object Linking
Libraries and plug-ins for GPU code

New Nsight™ Eclipse Edition
Develop, Debug, and Optimize... All in one tool!

GPUDirect™
RDMA between GPUs and PCIe devices
Hyper-Q

FERMI
1 Work Queue

KEPLER
32 Concurrent Work Queues
With Hyper-Q

Easier threaded parallelism
Multi-rank MPI parallelism
What is CUDA Dynamic Parallelism?

The ability for any GPU thread to launch a parallel GPU kernel

- Dynamically
- Simultaneously
- Independently

Fermi: Only CPU can generate GPU work

Kepler: GPU can generate work for itself

© 2012 NVIDIA
Dynamic Work Generation

Coarse grid

Higher Performance
Lower Accuracy

Fine grid

Lower Performance
Higher Accuracy

Dynamic grid

Target performance where accuracy is required
Familiar Syntax and Programming Model

```c
int main() {
    float *data;
    setup(data);
    A <<< ... >>> (data);
    B <<< ... >>> (data);
    C <<< ... >>> (data);
    cudaDeviceSynchronize();
    return 0;
}

__global__ void B(float *data)
{
    do_stuff(data);
    X <<< ... >>> (data);
    Y <<< ... >>> (data);
    Z <<< ... >>> (data);
    cudaDeviceSynchronize();
    do_more_stuff(data);
}
```
LU decomposition (Fermi)

```c

dgetrf(N, N) {
    for j=1 to N
        for i=1 to 64
            idamax<<<>>>(
            dswap<<<>>>(
            memcpy
            dscal<<<>>>(
            dger<<<>>>(
            next i
        memcpy
        dlaswap<<<>>>(
        dtrsm<<<>>>(
        dgemm<<<>>>(
        next j
}
```

CPU Code

GPU Code

LU decomposition (Kepler)

```c

dgetrf(N, N) {
    dgetrf<<<>>>(
    synchronize();
}

dgetrf(N, N) {
    for j=1 to N
        for i=1 to 64
            idamax<<<>>>(
            dswap<<<>>>(
            memcpy
            dscal<<<>>>(
            dger<<<>>>(
            next i
        dlaswap<<<>>>(
        dtrsm<<<>>>(
        dgemm<<<>>>(
        next j
    }
```

CPU Code

GPU Code

CPU is Free
Mapping Compute to the Problem

MYears: 4176.06
BODIES: 561750
Mapping Compute to the Problem
CUDA Dynamic Parallelism

Execution

GPU-Side Kernel Launch

Data-Dependent Execution
Recursive Parallel Algorithms
Batching to Help Fill GPU
Dynamic Load Balancing
Simplify CPU/GPU Divide
Library Calls from Kernels

Efficiency

Programmability
CUDA 4: Whole-Program Compilation & Linking

CUDA 4 required single source file for a single kernel
No linking external device code
CUDA 5: Separate Compilation & Linking

Separate compilation allows building independent object files

CUDA 5 can link multiple object files into one program
CUDA 5: Separate Compilation & Linking

Can also combine object files into static libraries
Link and externally call device code
Facilitates code reuse, reduces compile time
CUDA 5: Separate Compilation & Linking

Enables closed-source device libraries to call user-defined device callback functions
NVIDIA® Nsight™ Eclipse Edition

CUDA-Aware Editor
- Automated CPU to GPU code refactoring
- Semantic highlighting of CUDA code
- Integrated code samples & docs

Nsight Debugger
- Simultaneously debug of CPU and GPU
- Inspect variables across CUDA threads
- Use breakpoints & single-step debugging

Nsight Profiler
- Quickly identifies performance issues
- Integrated expert system
- Source line correlation

Available for Linux and Mac OS

© 2012 NVIDIA
NVIDIA® Nsight™ Eclipse Edition

CUDA aware editor

- Integrated CUDA samples makes it quick and easy to get started
- Easily port CPU loops to CUDA kernels with automatic code refactoring
- Semantic highlighting of CUDA code makes it easy to differentiate GPU code from CPU code
- Generate code faster with CUDA aware auto code completion and inline help
- Hyperlink navigation enables faster code browsing
- Supports automatic makefile generation

© 2012 NVIDIA
NVIDIA® Nsight™ Eclipse Edition

Nsight Debugger

- Seamless and simultaneous debugging of both CPU and GPU code
- View program variables across several CUDA threads
- Examine execution state and mapping of the kernels and GPUs
- View, navigate and filter to selectively track execution across threads
- Set breakpoints and single-step execution at both source-code and assembly levels
- Includes CUDA-MEMCHECK to help detect memory errors
Easily identify performance bottlenecks using a unified CPU and GPU trace of application activity

- Expert analysis system pin-points potential optimization opportunities

- Highlights potential performance problems at specific source-lines within application kernels

- Close integration with Nsight editor and builder for fast edit-build-profile optimization cycle

- Integrates with the new nvprof command-line profiler to enable visualization of profile data collected on headless compute nodes
CUDA on Mac!
NVIDIA GPUDirect™ now supports RDMA

RDMA: Remote Direct Memory Access between any GPUs in your cluster
CUDA Compiler Contributed to Open Source LLVM

Developers want to build front-ends for Java, Python, R, DSLs

Target other processors like ARM, FPGA, GPUs, x86
Try out CUDA 5

CUDA 5.0 Release Candidate
- Available early next week!
- Full support for all CUDA 5.0 features
- Use GPU linking and NSIGHT EE—both work with Fermi & GK10x
- Peruse early documentation and header files for GK110 features
  - SM 3.5 support and Dynamic Parallelism
- Provide feedback to NVIDIA via CUDA Forums and CUDA_RegDev@nvidia.com

CUDA 5.0 Preview (alpha)
- Become a registered developer and download CUDA 5.0 preview
How to get started

www.nvidia.com/cudazone

www.nvidia.com/getcuda
GTC 2013 | March 18-21 | San Jose, CA
The Smartest People. The Best Ideas. The Biggest Opportunities.

Opportunities for Participation:

SPEAK - Showcase your work among the elite of graphics computing
- Call for Sessions: August 2012
- Call for Posters: October 2012

REGISTER - learn from the experts and network with your peers
- Use promo code GM10SIGG for a 10% discount

SPONSOR - Reach influential IT decision-makers

Learn more at www.gputechconf.com