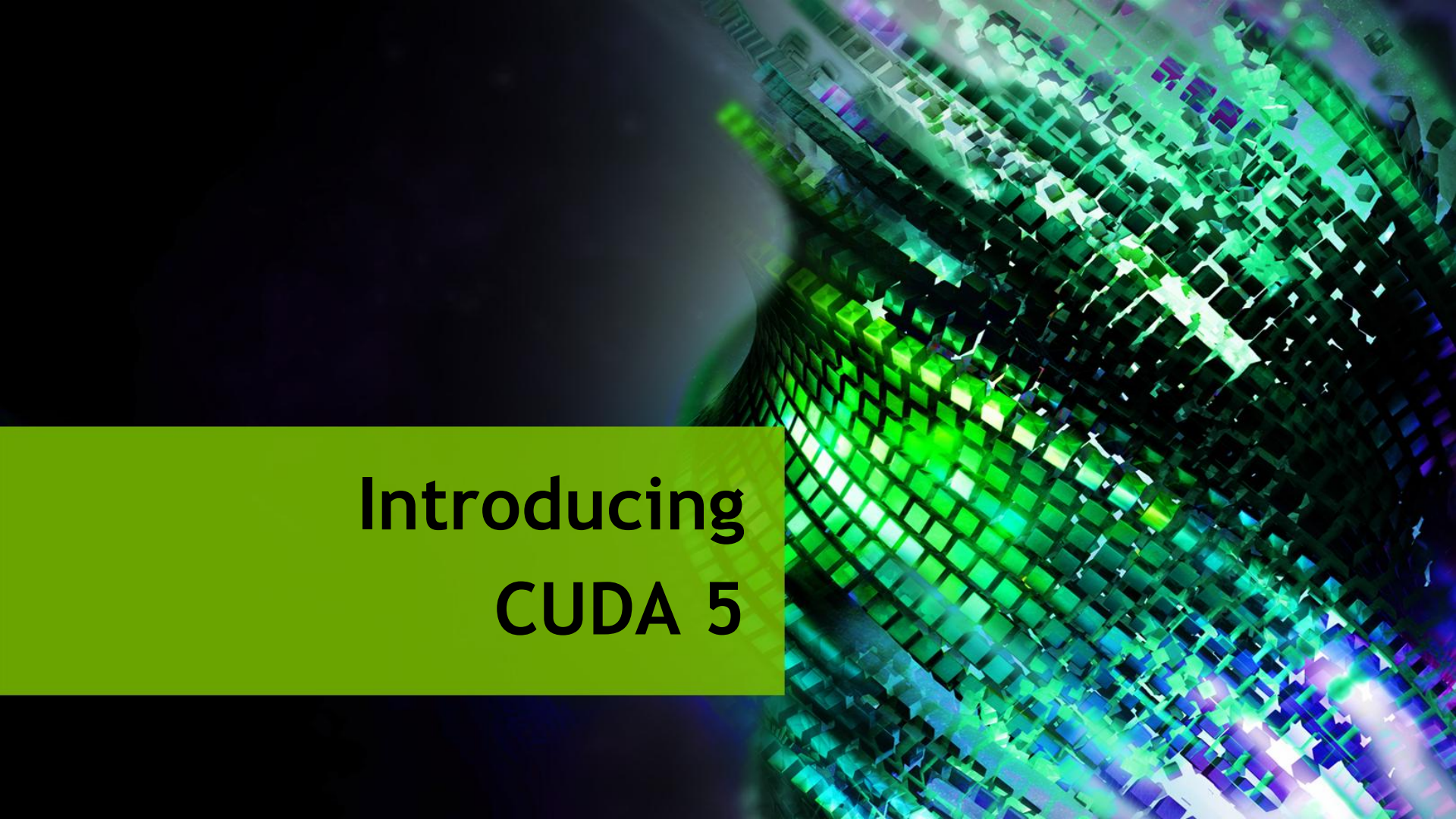

Introducing
CUDA 5

CUDA 5

Application Acceleration Made Easier

Dynamic Parallelism

Spawn new parallel work from within GPU code on K20

GPU Callable Libraries

Libraries and plug-ins for GPU code

GPUDirect™

RDMA between GPUs and PCIe devices

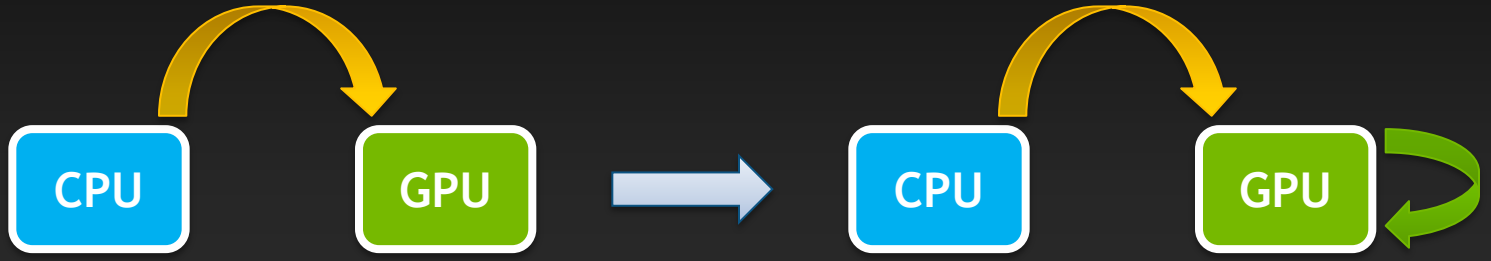
New Nsight™ Eclipse Edition

Develop, Debug, and Optimize… all in one tool!

What is Dynamic Parallelism?

The ability to launch new kernels from the GPU

- Dynamically - based on run-time data
- Simultaneously - from multiple threads at once
- Independently - each thread can launch a different grid

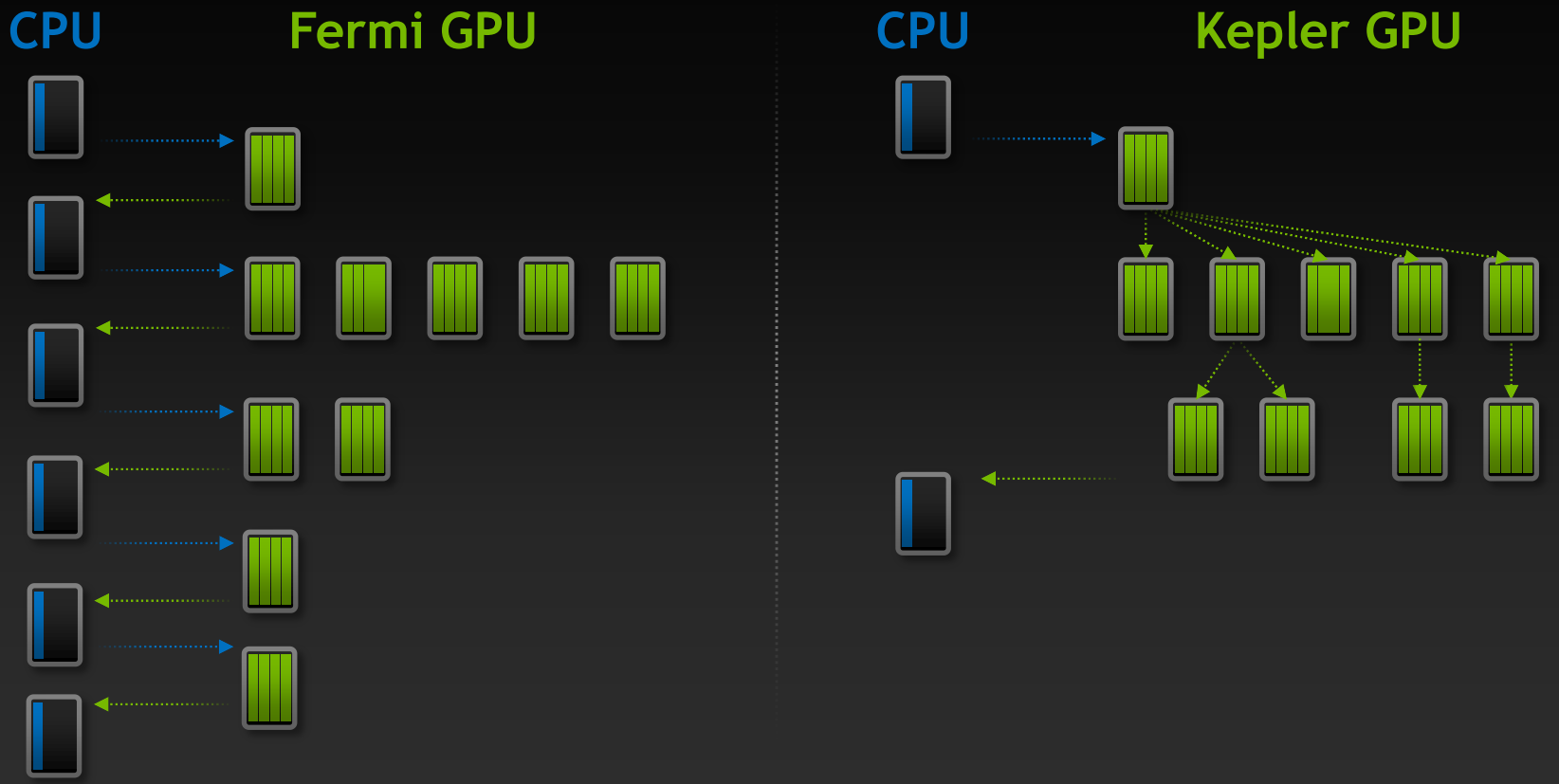


Fermi: Only CPU can generate GPU work

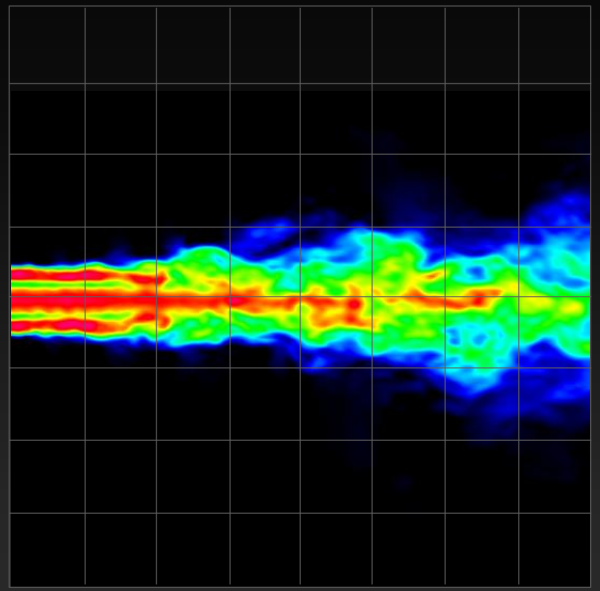
Kepler: GPU can generate work for itself

Dynamic Parallelism

Accelerate More of Your Application on the GPU



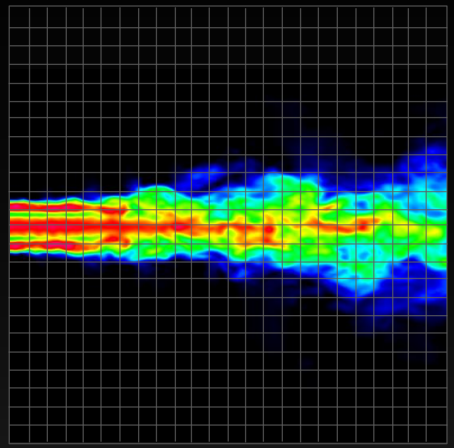
Dynamic Work Generation



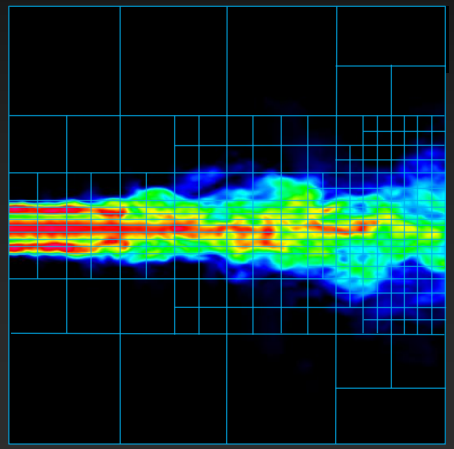
Initial Grid

Statically assign conservative worst-case grid

Fixed Grid



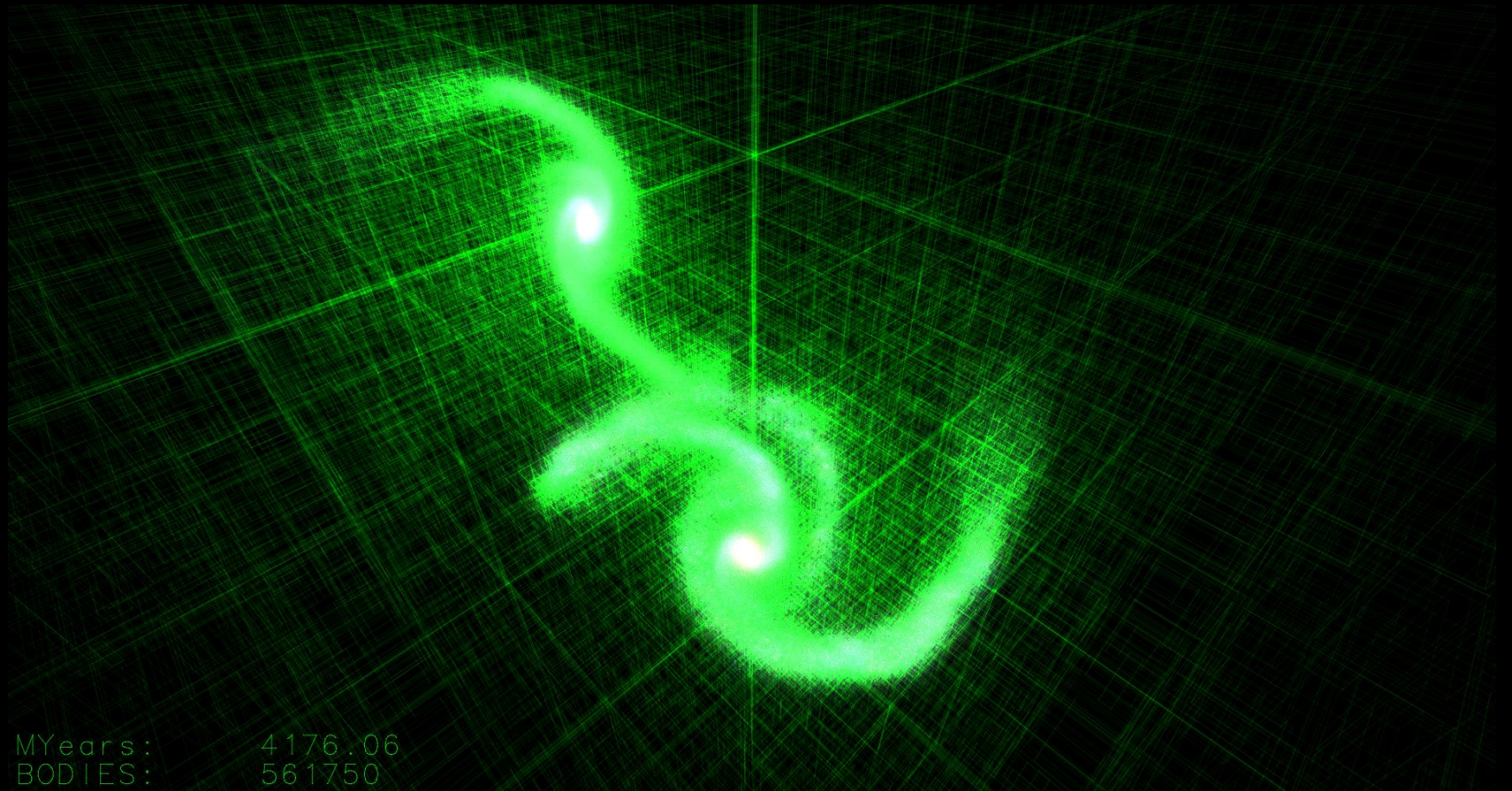
Dynamically assign performance where accuracy is required



Dynamic Grid

Mapping Compute to the Problem

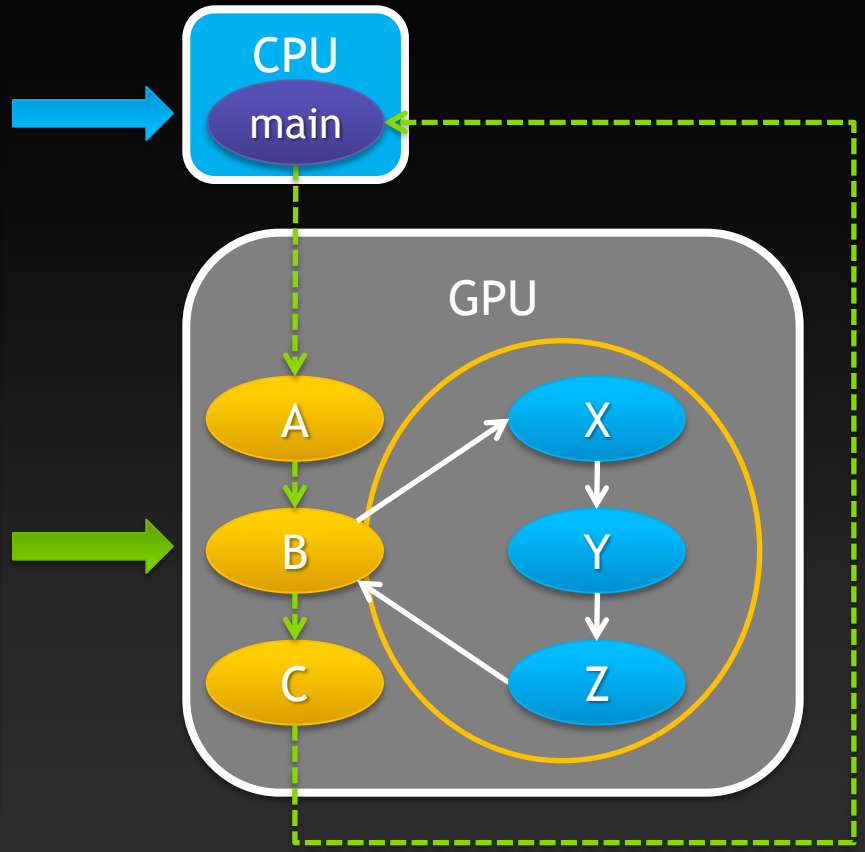
Mapping Compute to the Problem



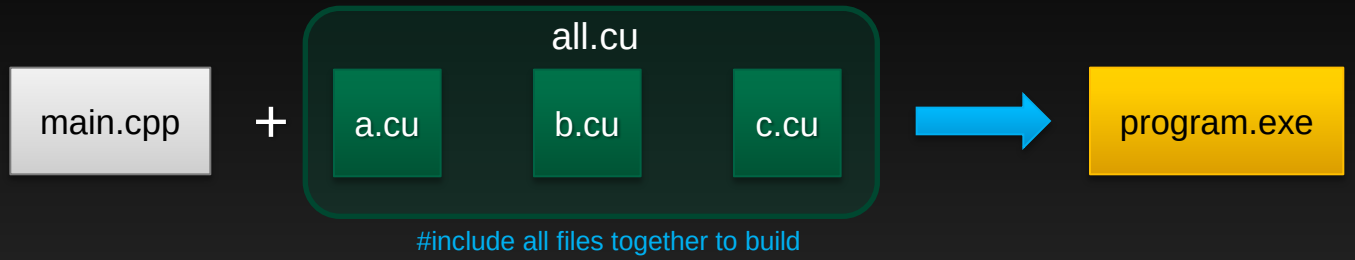
Familiar Syntax and Programming Model

```
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);
}
```



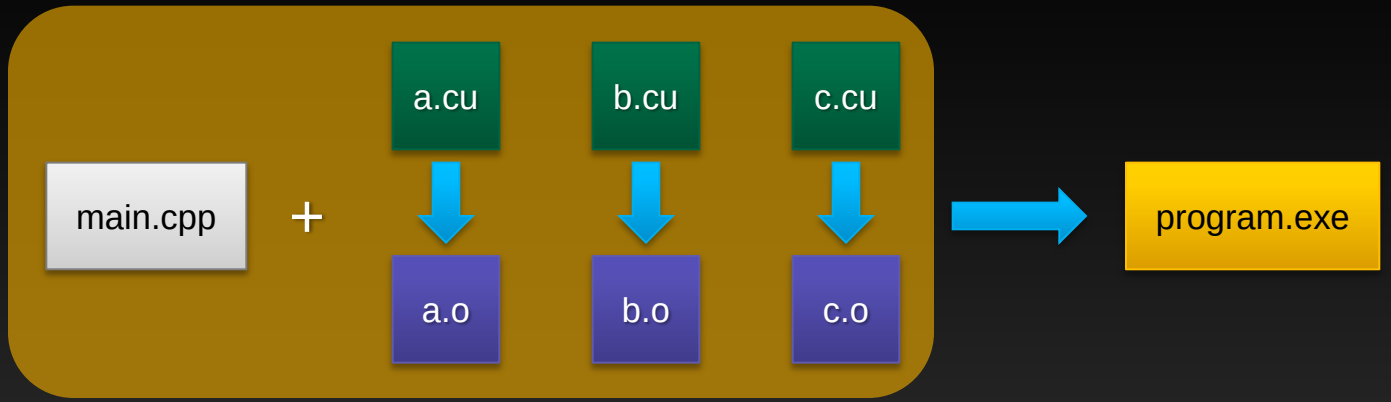
Before CUDA 5: Whole-Program Compilation



Earlier CUDA releases required a single source file for each kernel

Linking with external code was not supported

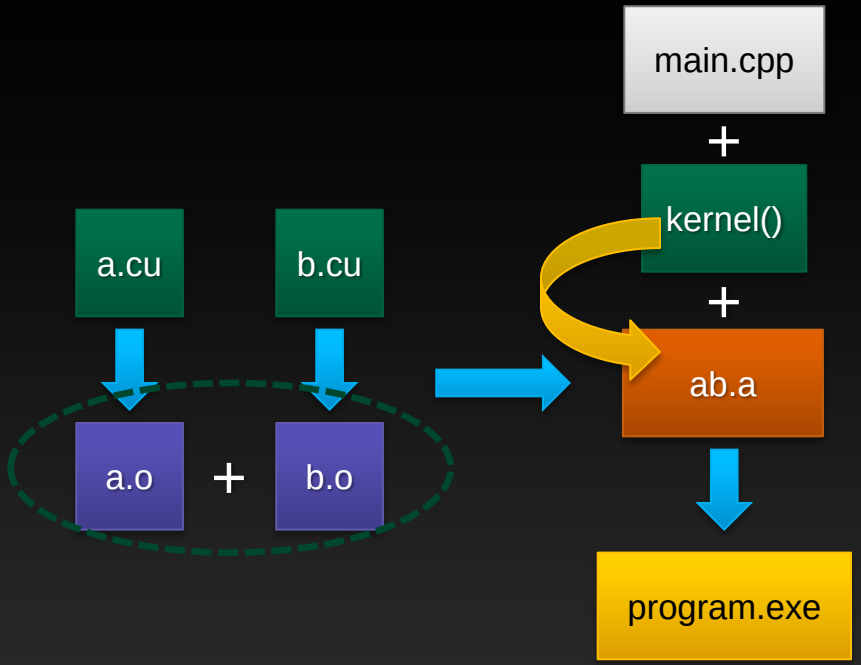
CUDA 5: Separate Compilation & Linking



Separate compilation allows building independent object files

CUDA 5 can link multiple object files into one program

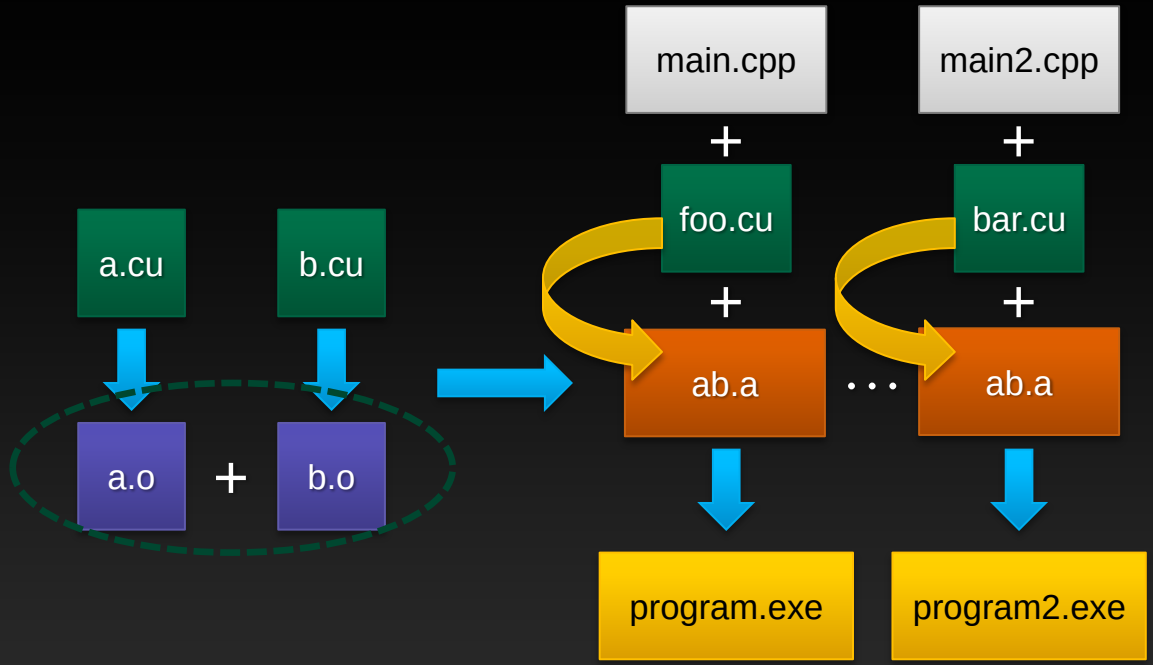
CUDA 5: GPU Callable Libraries



Can combine object files into static libraries

Link and externally call *device* code

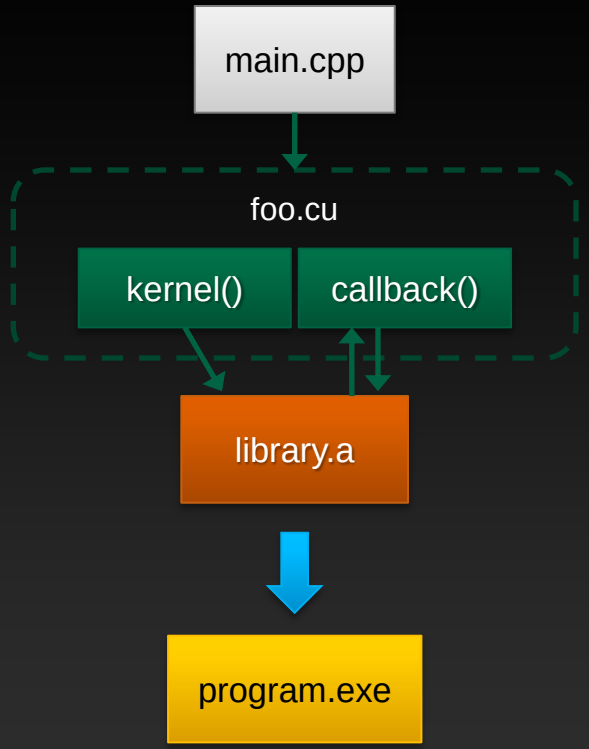
CUDA 5: GPU Callable Libraries



Facilitates code reuse, reduces compile time

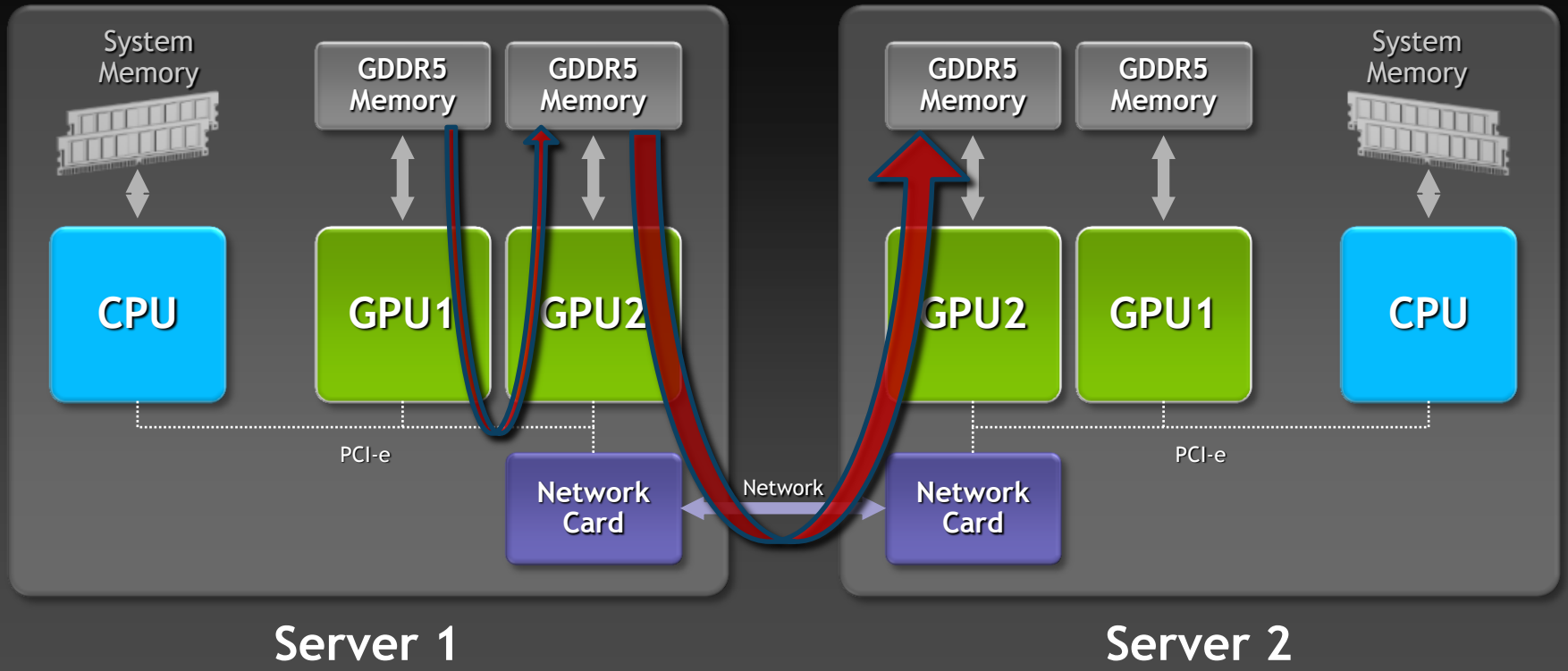
CUDA 5: Callbacks

Enables closed-source device libraries to call user-defined device callback functions

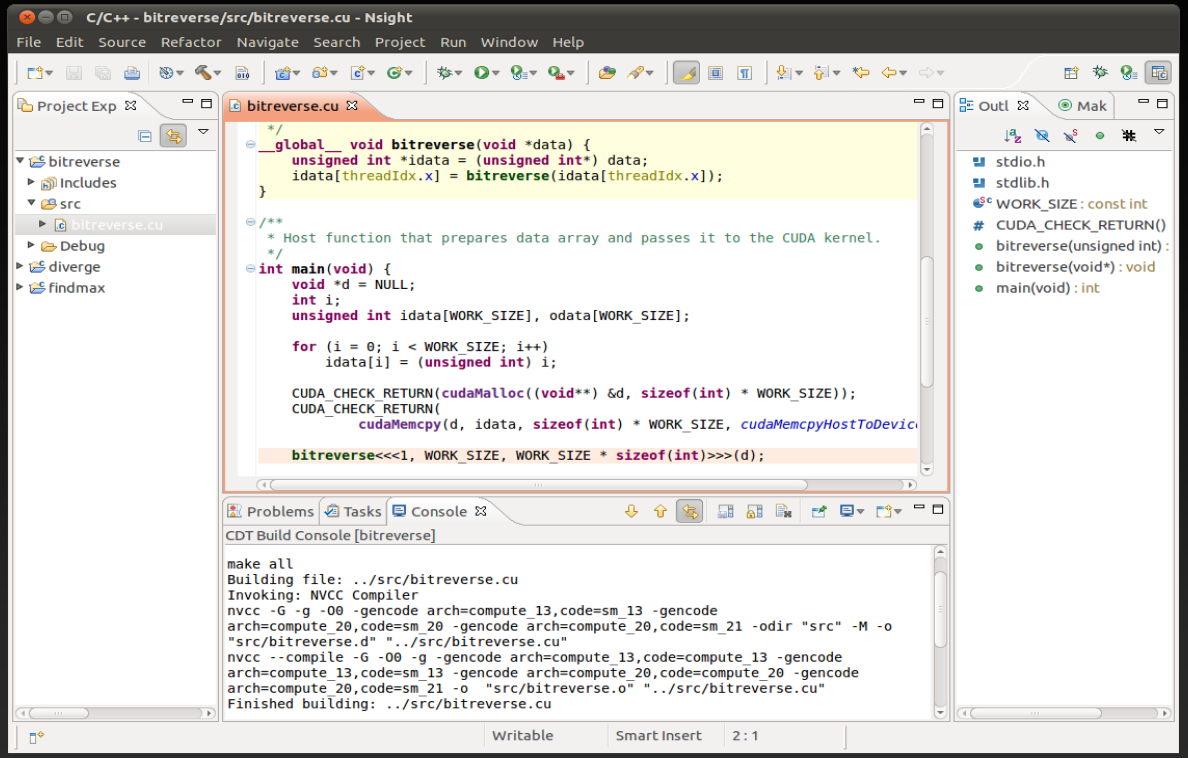


NVIDIA® GPUDirect™ Support for RDMA

Direct Communication Between GPUs and PCIe devices



NVIDIA® Nsight,™ Eclipse Edition

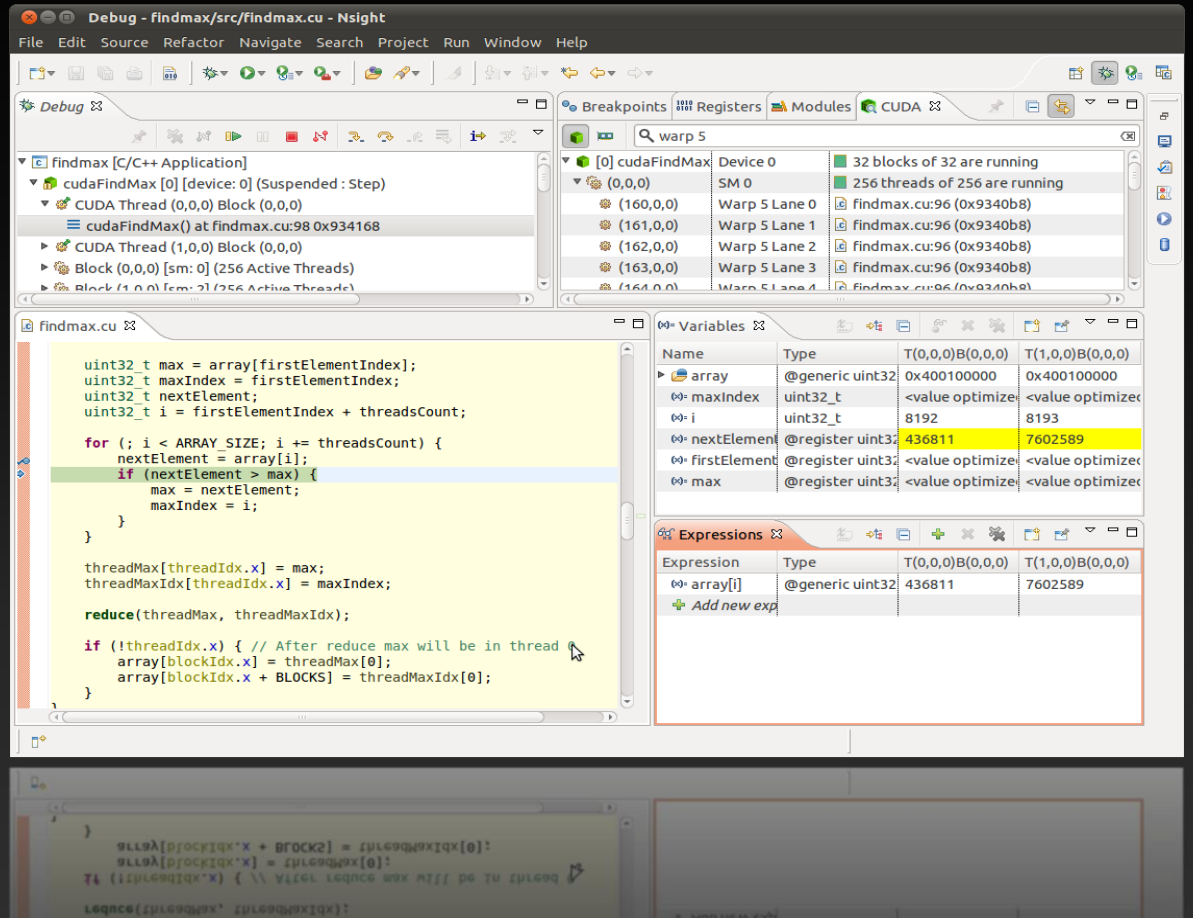


CUDA aware editor

- 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

www.nvidia.com/nsight

NVIDIA® Nsight,™ Eclipse Edition

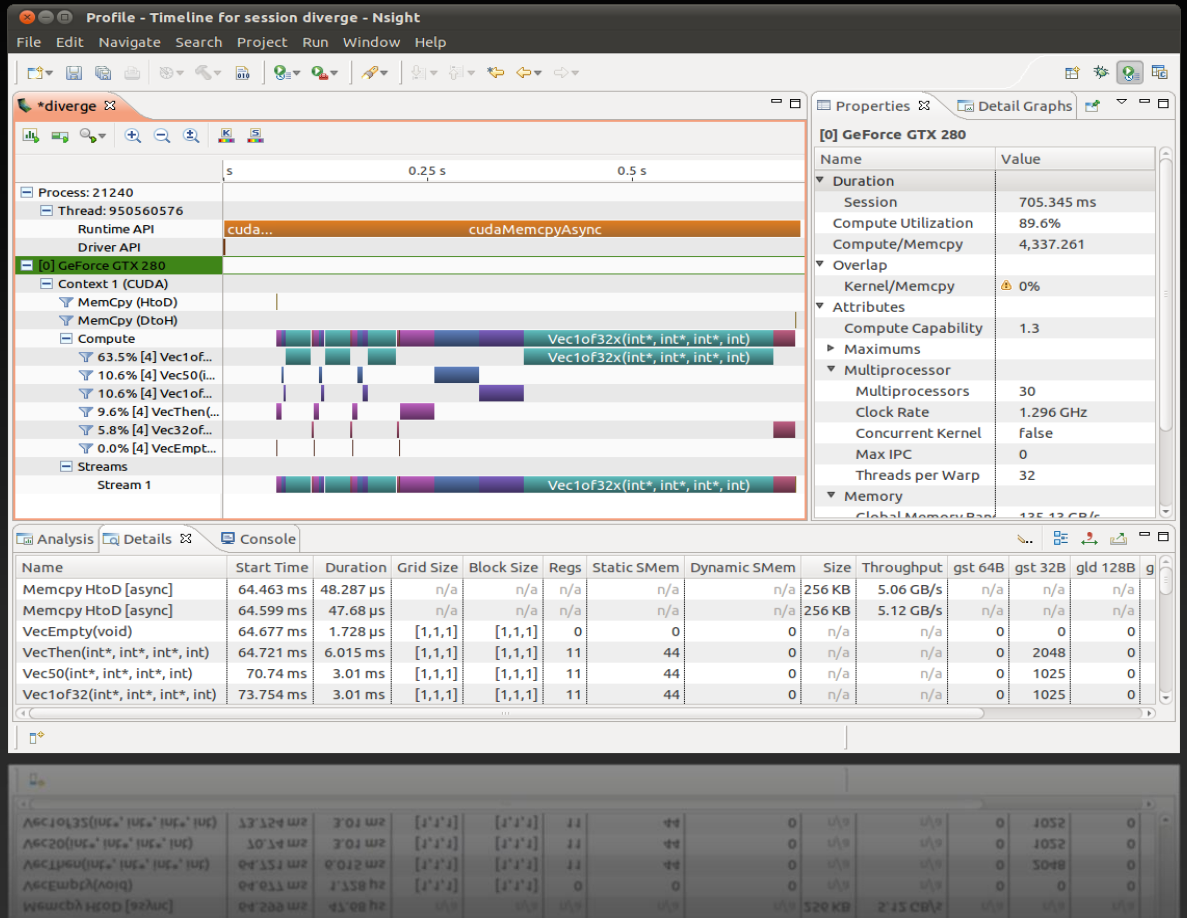


Nsight Debugger

- 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

www.nvidia.com/nsight

NVIDIA® Nsight,™ Eclipse Edition



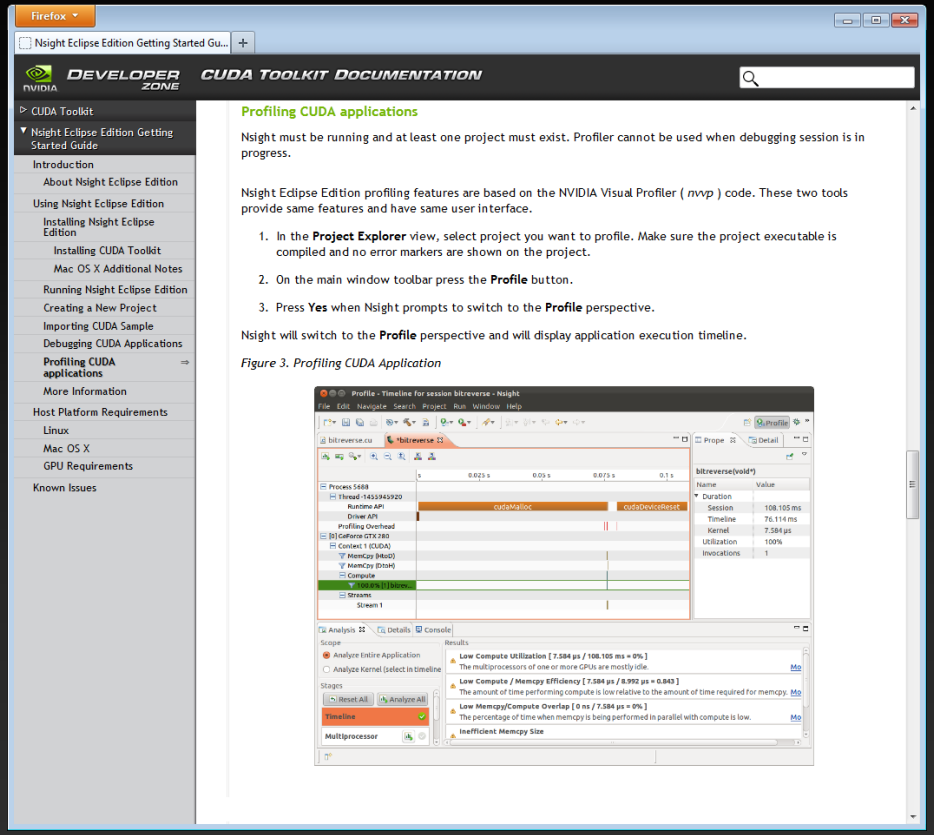
Nsight Profiler

- 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

www.nvidia.com/nsight

New Online CUDA Resource Center

All the Latest, All in One Place



- Programming Guides
- API Reference
- Library Manuals
- Code Samples
- Tools Manuals
- Platform Specs

Over 1600 Files!

http://docs.nvidia.com

Summary

1. Dynamic Parallelism accelerates more of your application on Kepler GPUs
2. GPU Callable Libraries enable 3rd party ecosystem and customization via callbacks and plug-ins
3. GPUDirect support for RDMA enables high performance, low-latency communication between GPUs and other PCIe devices
4. New Nsight™ Eclipse Edition increases programmer productivity for world's most popular parallel programming platform

www.nvidia.com/getcuda

Thanks!