CUDA
5 YEARS

CUDA 4.1

Application Acceleration Made Easier

New LLVM-Based Compiler

Delivers +10% performance for many applications

1,000+ New Image Processing Functions

"Drop-in" acceleration with NPP library

Re-Designed Visual Profiler

Automated Analysis & Integrated Expert Guidance

New LLVM-based CUDA Compiler

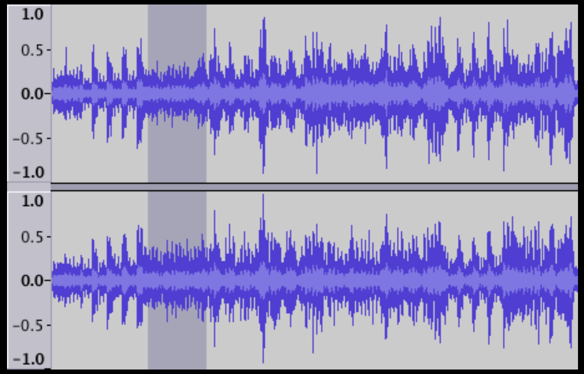
- Delivers up to 10% faster application performance
- Faster compilation for increased developer productivity
- Modern compiler with broad support
  - Will bring more languages to the GPU
  - Easier to support CUDA to more platforms

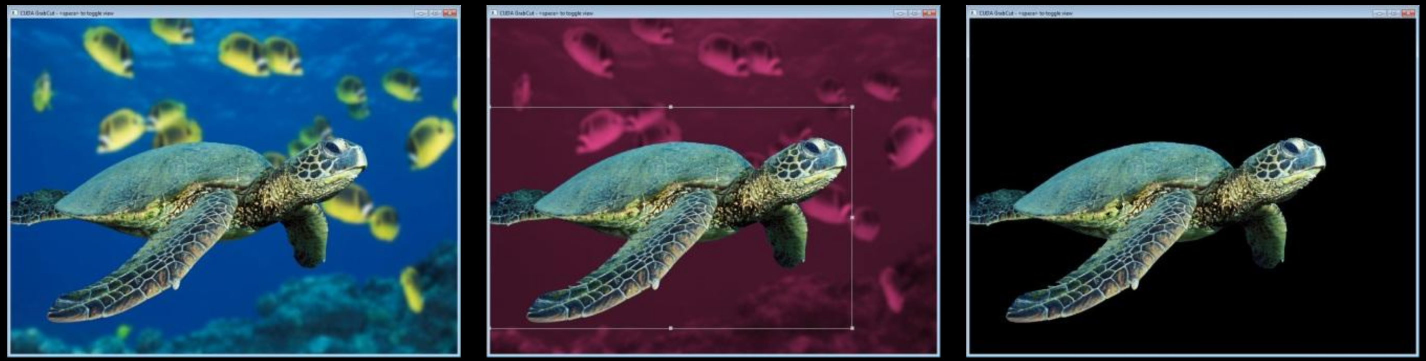
1000+ New Imaging Functions in NPP 4.1

- NVIDIA Performance Primitives (NPP) library includes over 2200 GPU-accelerated functions for image & signal processing

  Arithmetic, Logic, Conversions, Filters, Statistics, etc.

- Up to 40x faster performance than Intel IPP

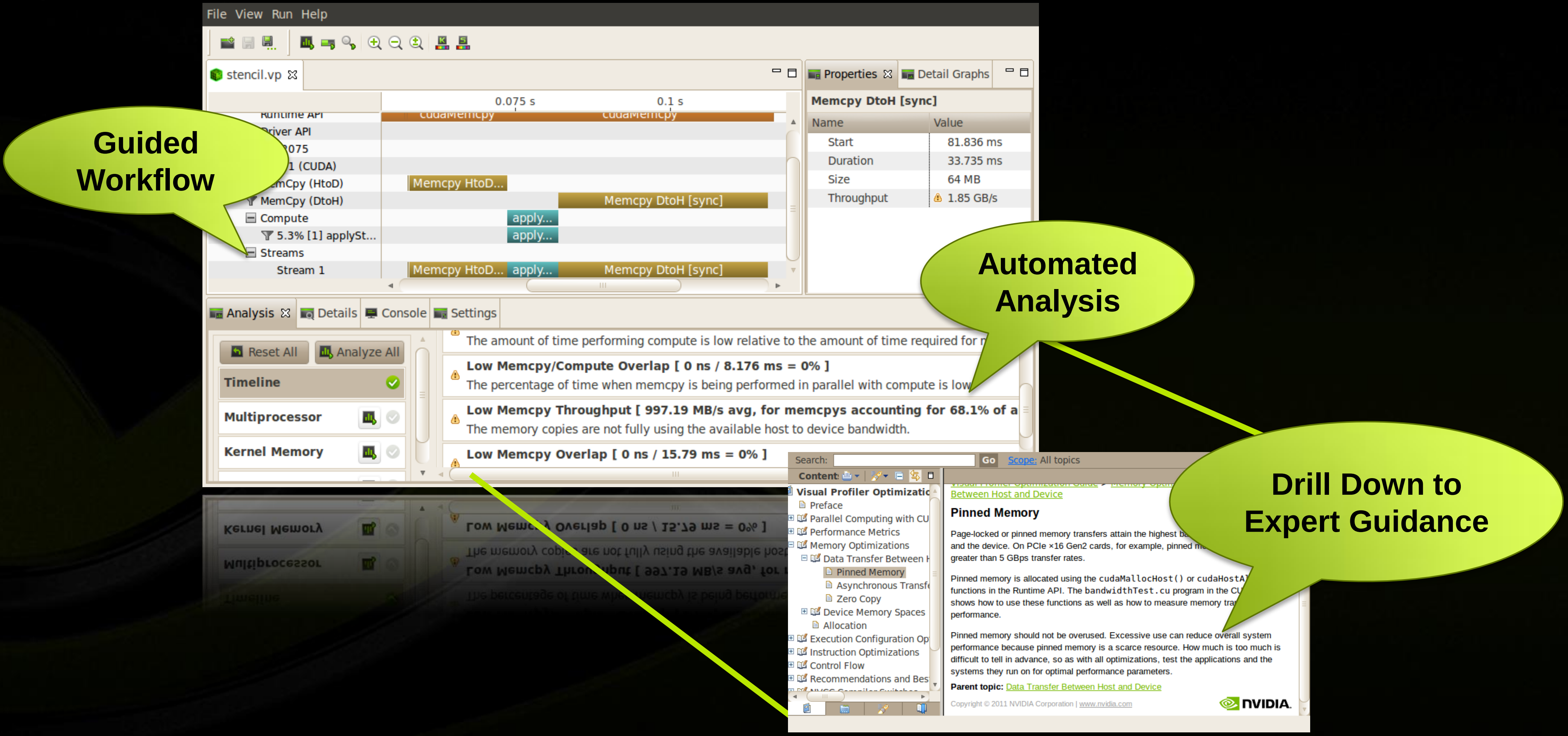




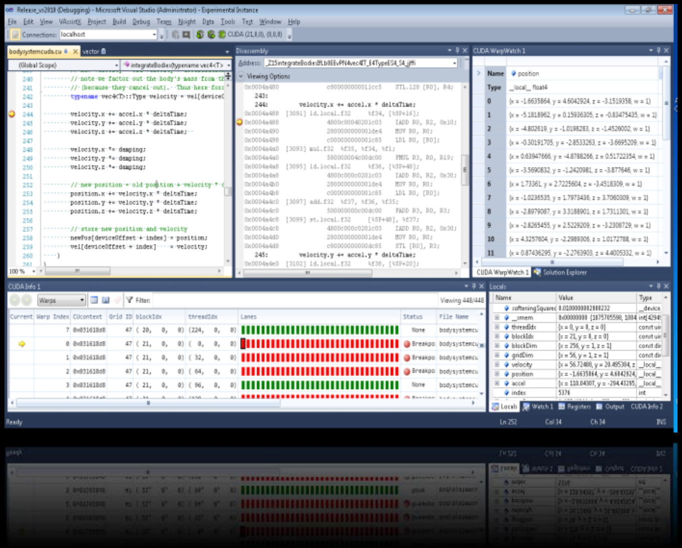
http://developer.nvidia.com/content/graphcuts-using-npp

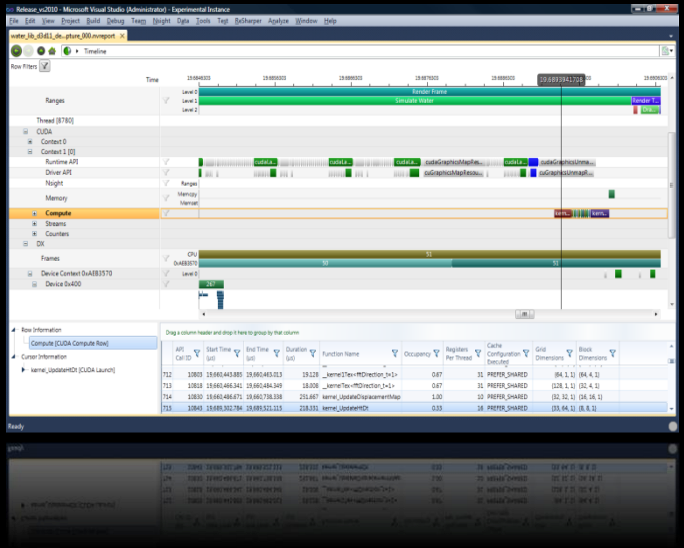
* NPP 4.1, NVIDIA C2050 (Fermi)
* IPP 6.1, Dual Socket Core™ i7 920 @ 2.67GHz

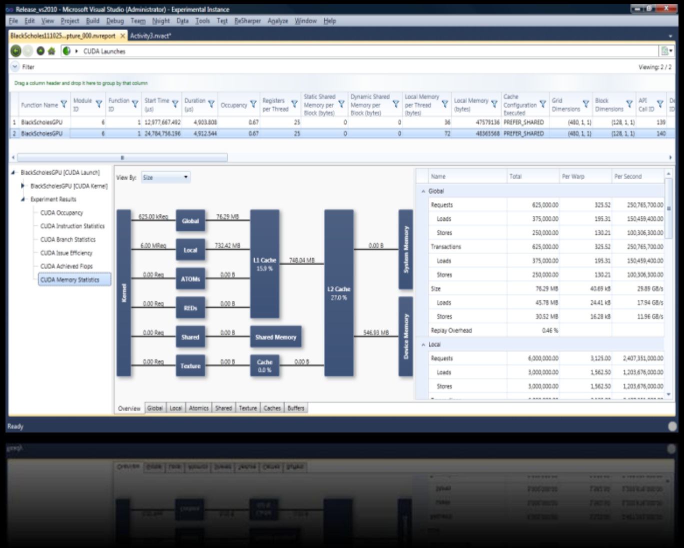
Re-designed Visual Profiler



NVIDIA Parallel Nsight™ 2.1 for Visual Studio







CUDA Debugger

- Debug CUDA kernels directly on GPU hardware
- Examine thousands of threads executing in parallel
- Use on-target conditional breakpoints to locate errors

CUDA Memory Checker

- Enables precise error detection

System Trace

- Review CUDA activities across CPU and GPU
- Perform deep kernel analysis to detect factors limiting maximum performance

CUDA Profiler

- Advanced experiments to measure memory utilization, instruction throughput and stalls

GPU-Aware MPI Libraries

Integrated Support for GPU Computing

OPENFABRICS ALLIANCE

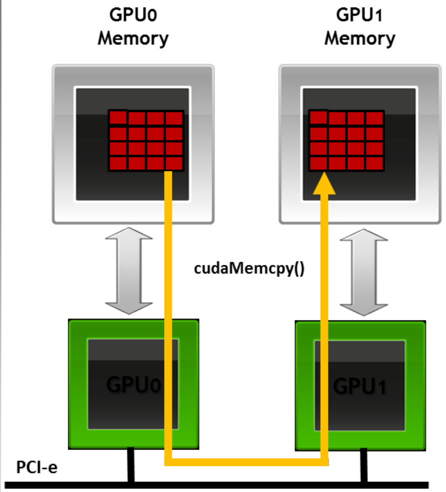
As of OFED 1.5.2

MVAPICH

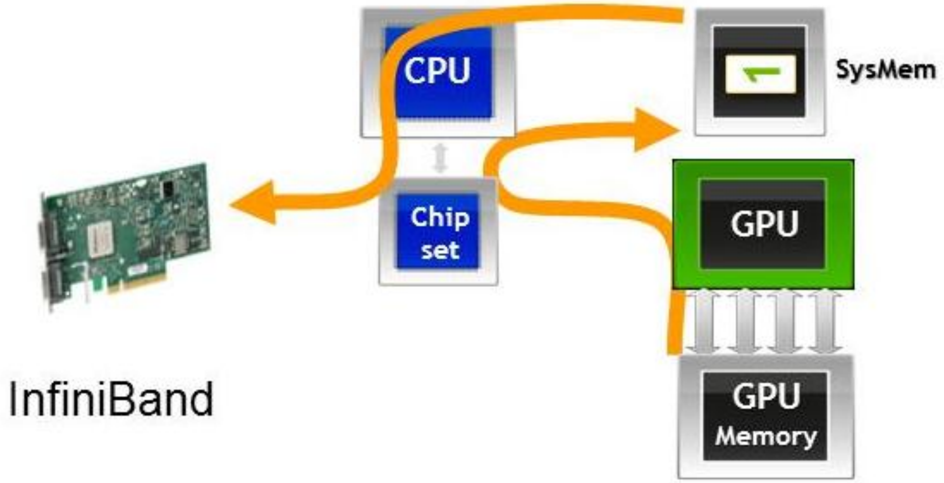
Pre-Release
Announced at SC'11

Platform Computing

Platform MPI
Beta announced at SC'11



Peer-to-Peer Transfers



Accelerated Communication

http://developer.nvidia.com/gpudirect

CUDA 4.1 Highlights

Advanced Application Development

- New LLVM-based compiler
- 3D surfaces & cube maps
- Peer-to-Peer between processes
- GPU reset with nvidia-smi
- New GrabCut sample shows interactive foreground extraction
- New code samples for optical flow, volume filtering and more…

GPU-Accelerated Libraries

- 1000+ new imaging functions
- Tri-diagonal solver 10x faster vs. MKL
- MRG32k3a & MTGP11213 RNGs
- New Bessell functions in Math lib
- 2x faster matrix-vector w/ HYB-ELL
- Boost-style placeholders in Thrust
- Batched GEMM for small matrices

New & Improved Developer Tools

- Re-Designed Visual Profiler
- Parallel Nsight 2.1
- Multi-context debugging
- assert() in device code
- Enhanced CUDA-MEMCHECK

NVIDIA CUDA Platform

CUDA 4.1 Highlights

Platform

New LLVM-based Compiler

Hardware Support
ECC Memory
Double Precision
Native 64-bit Architecture
GPUDirect™ Communication
Concurrent Kernel Execution
Dual Copy Engines
Multi-GPU support
6GB per GPU supported

Operating System Support
MS Windows 32/64
Linux 32/64 support
Mac OSX 32/64 support

Cluster Management
Tesla Compute Cluster (TCC)
Unified Virtual Addressing
Graphics Interoperability

Programming Model

P2P Between Processes
3D Surfaces & Cubemaps

C support
- NVIDIA C Compiler
- CUDA C Parallel Extensions
- Function Pointers
- Recursion
- Atomics
- malloc/free

C++ support
- Classes/Objects
- new/delete
- Class Inheritance
- Polymorphism
- Operator Overloading
- Class Templates
- Function Templates
- Virtual Functions
- Virtual Base Classes
- Namespaces

Fortran also available

Libraries

1000+ New Imaging Functions
New Tri-diagonal solver

CUDA Toolkit Libraries
Complete math.h
Complete BLAS Library (1, 2 and 3)
Sparse Matrix Math Library
RNG Library
FFT Library (1D, 2D and 3D)
Thrust Template Library
Image Processing Library (NPP)
Video Processing Library (NPP)
Video Codec Libraries

Additional Libraries
- CULA Tools
- MAGMA
- IMSL
- VSIPL
- CUSP

Tools

New Visual Profiler
Parallel Nsight 2.1

NVIDIA Developer Tools
Parallel Nsight 1.0 IDE
cuda-gdb Debugger with multi-GPU
CUDA/OpenCL Visual Profiler
CUDA Memory Checker
CUDA C SDK
CUDA Disassembler

3rd Party Developer Tools
Allinea DDT, Totalview
PAPI, TAU, Vampir

Languages & APIs
CUDA Fortran
CUDA-x86 for CPUs
OpenACC
PGI Accelerator for C / Fortran
CAPS HMPP
Python, C#,
DirectCompute, OpenCL
Many more…

CUDA Libraries Performance Report

CUDA Math Libraries

High performance math routines for your applications:

- cuFFT - Fast Fourier Transforms Library
- cuBLAS - Complete BLAS Library
- cuSPARSE - Sparse Matrix Library
- cuRAND - Random Number Generation (RNG) Library
- NPP - Performance Primitives for Image & Video Processing
- Thrust - Templated Parallel Algorithms & Data Structures
- math.h - C99 floating-point Library

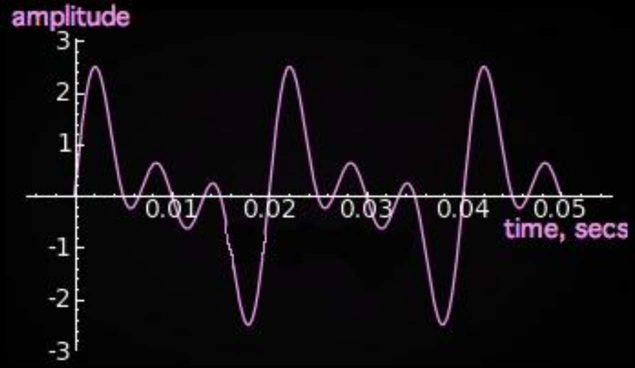
Included in the CUDA Toolkit   Free download @ www.nvidia.com/getcuda

More information on CUDA libraries:

http://www.nvidia.com/object/gtc2010-presentation-archive.html#session2216

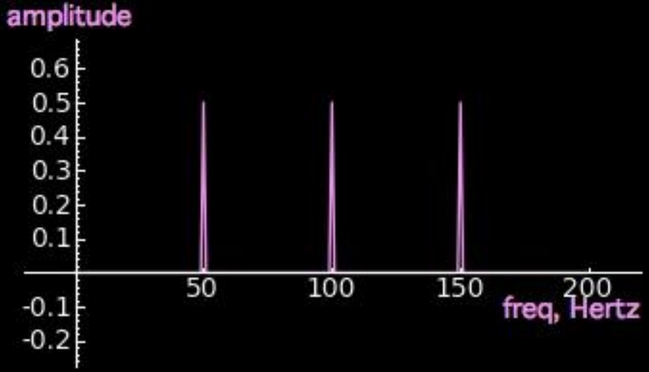
cuFFT: Multi-dimensional FFTs

- New in CUDA 4.1
  - Flexible input & output data layouts for all transform types
    - Similar to the FFTW “Advanced Interface”
    - Eliminates extra data transposes and copies
  - API is now thread-safe & callable from multiple host threads
  - Restructured documentation to clarify data layouts



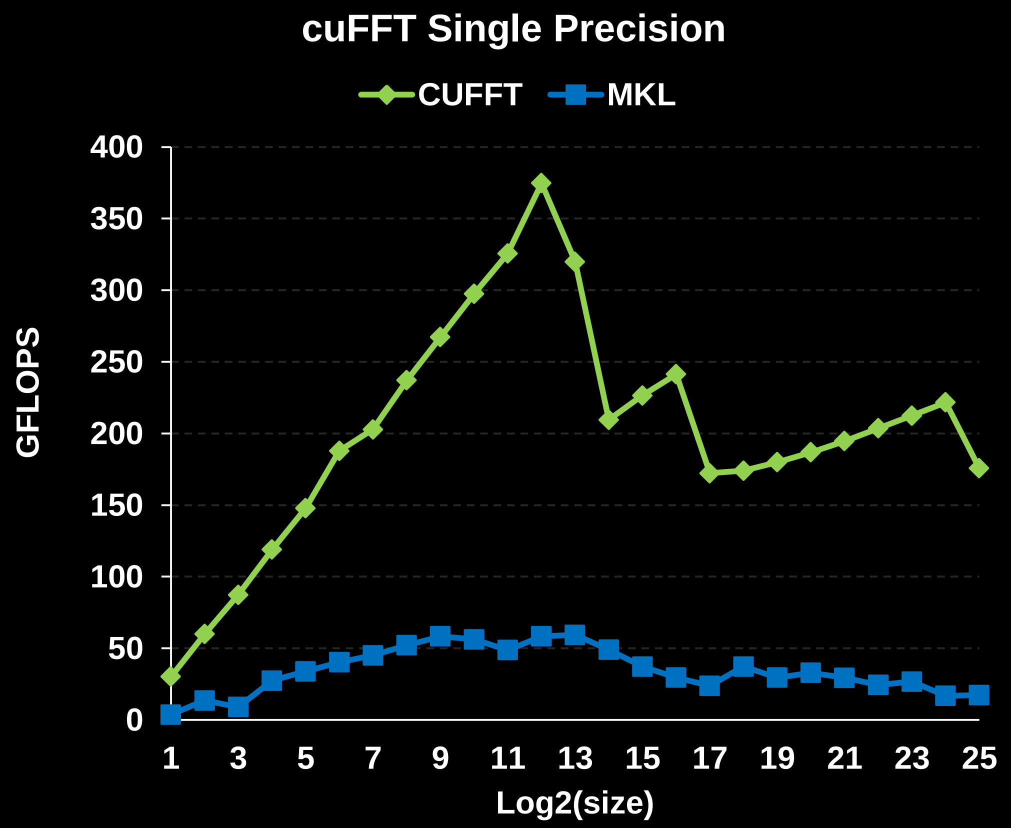
$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi(x\frac{n}{N})}$$

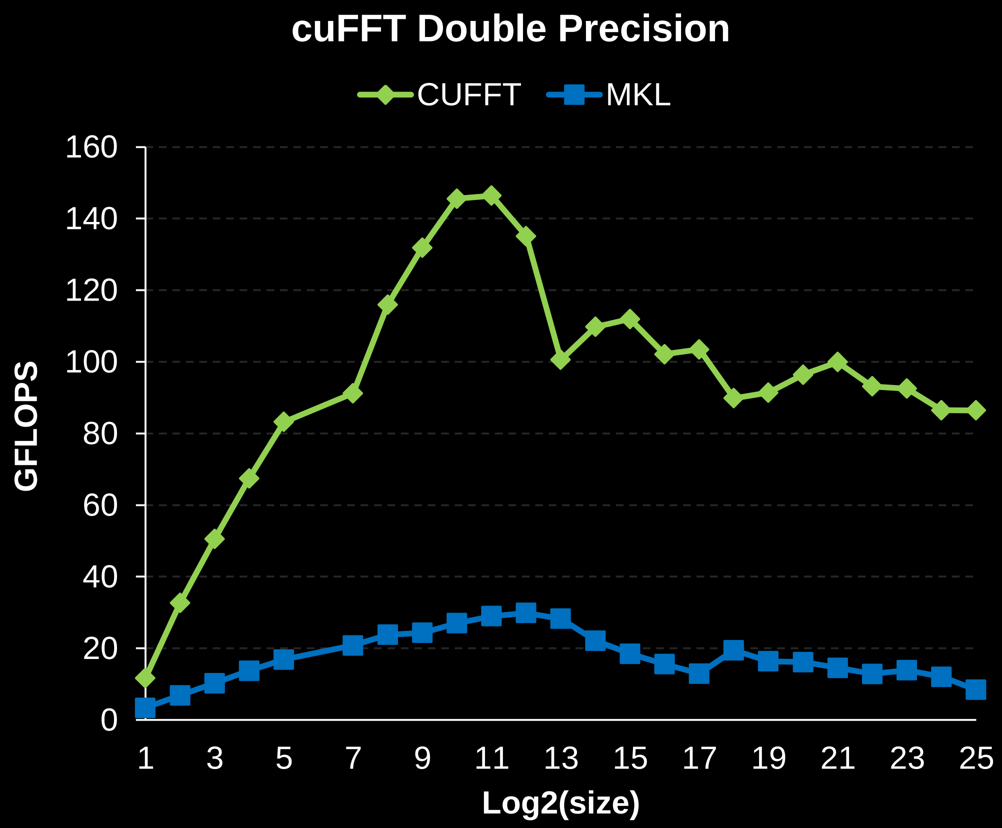
$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) e^{j2\pi(x\frac{n}{N})}$$



FFTs up to 10x Faster than MKL

1D used in audio processing and as a foundation for 2D and 3D FFTs

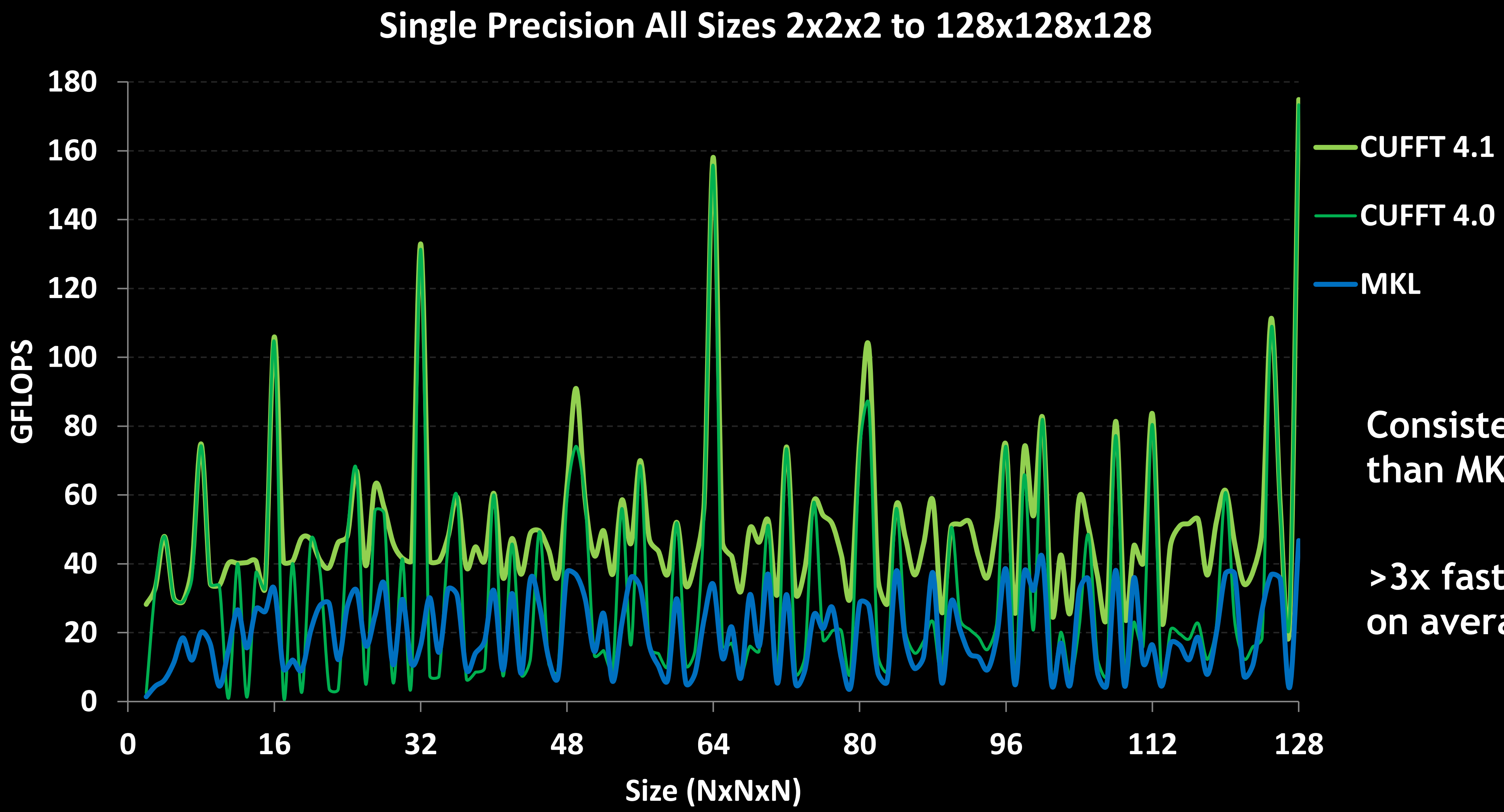




- Measured on sizes that are exactly powers-of-2
- cuFFT 4.1 on Tesla M2090, ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Performance may vary based on OS version and motherboard configuration

CUDA 4.1 optimizes 3D transforms



Consistently faster than MKL

>3x faster than 4.0 on average

- cuFFT 4.1 on Tesla M2090, ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Performance may vary based on OS version and motherboard configuration

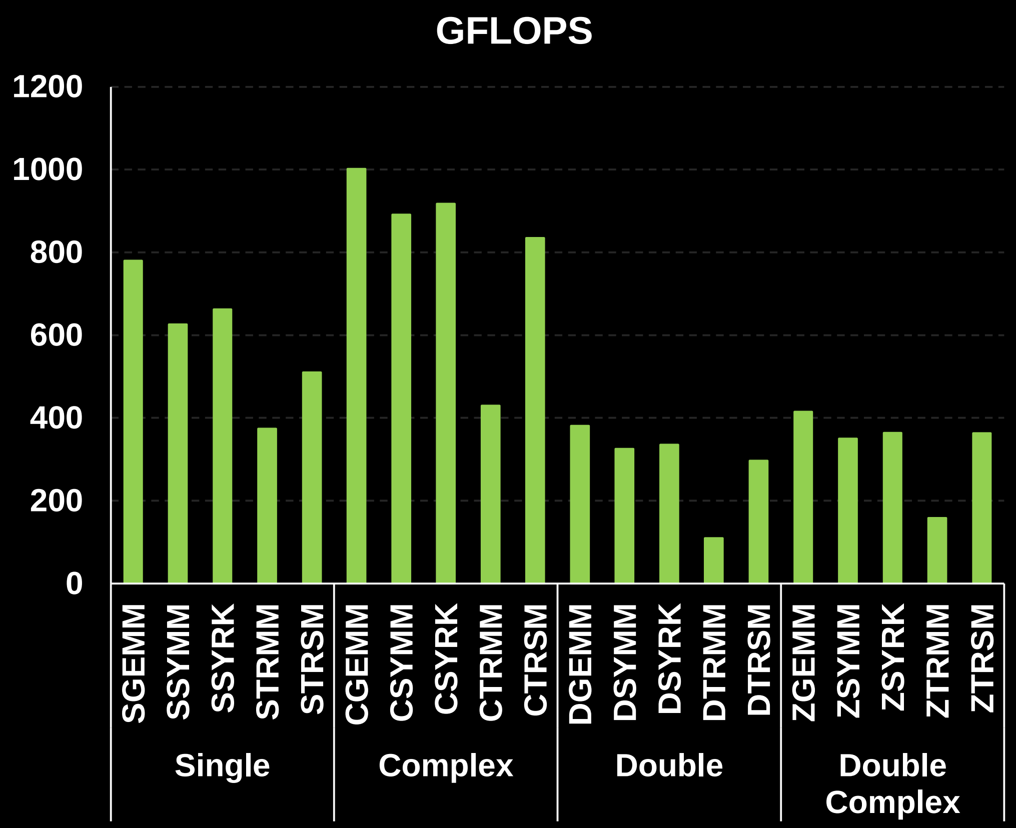
cuBLAS: Dense Linear Algebra on GPUs

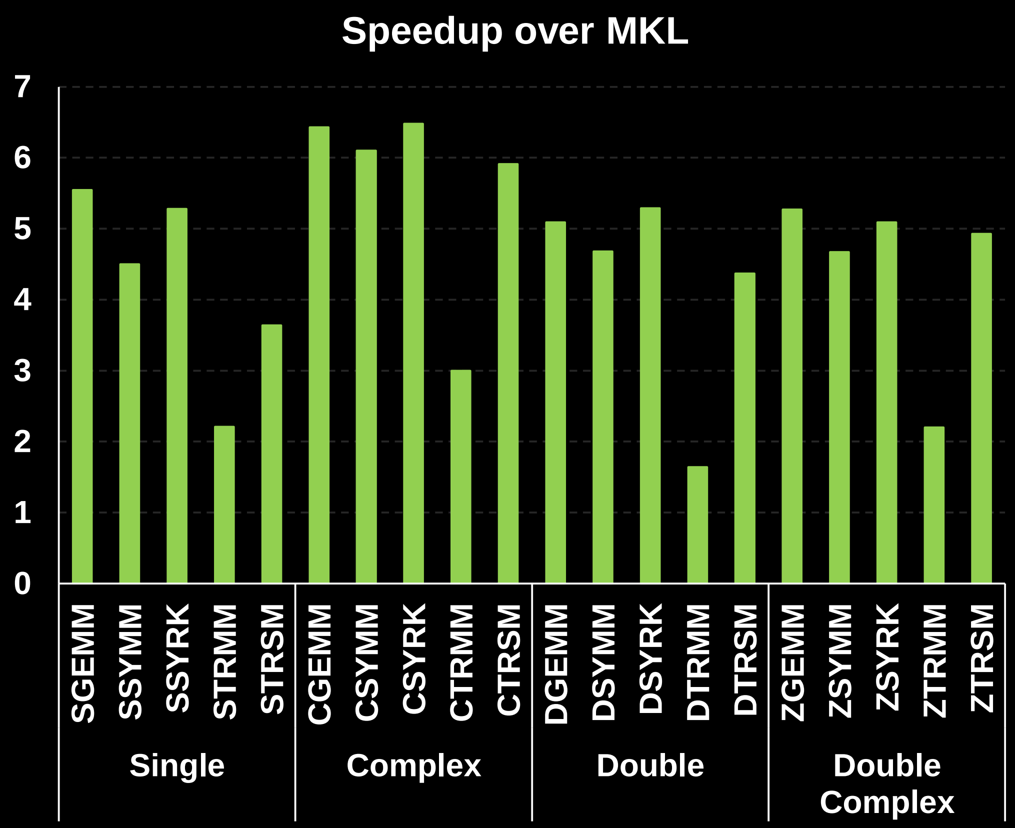
- Complete BLAS implementation plus useful extensions
  - Supports all 152 standard routines for single, double, complex, and double complex

- New in CUDA 4.1
  - New batched GEMM API provides >4x speedup over MKL
    - Useful for batches of 100+ small matrices from 4x4 to 128x128
  - 5%-10% performance improvement to large GEMMs

cuBLAS Level 3 Performance

Up to 1 TFLOPS sustained performance and >6x speedup over Intel MKL

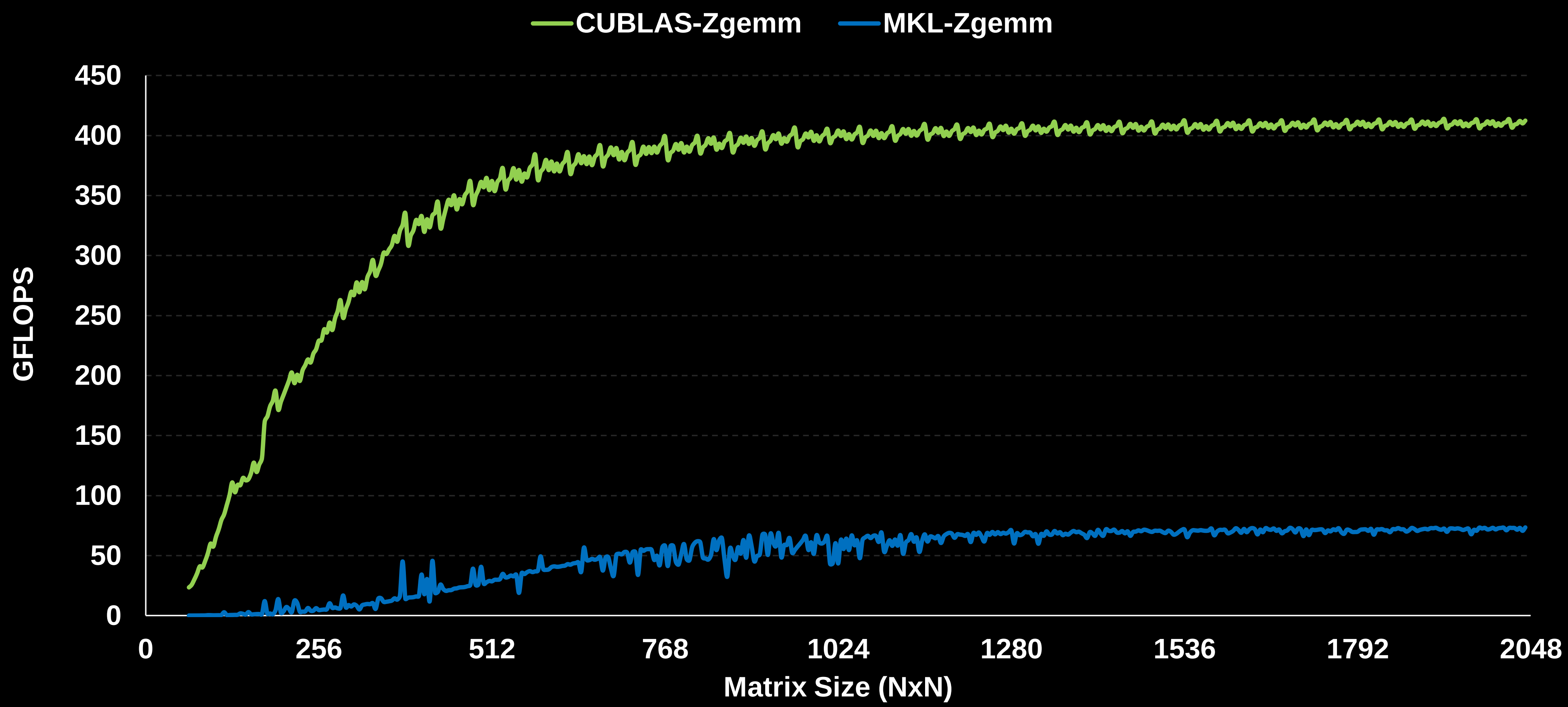




- 4Kx4K matrix size
- cuBLAS 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Performance may vary based on OS version and motherboard configuration

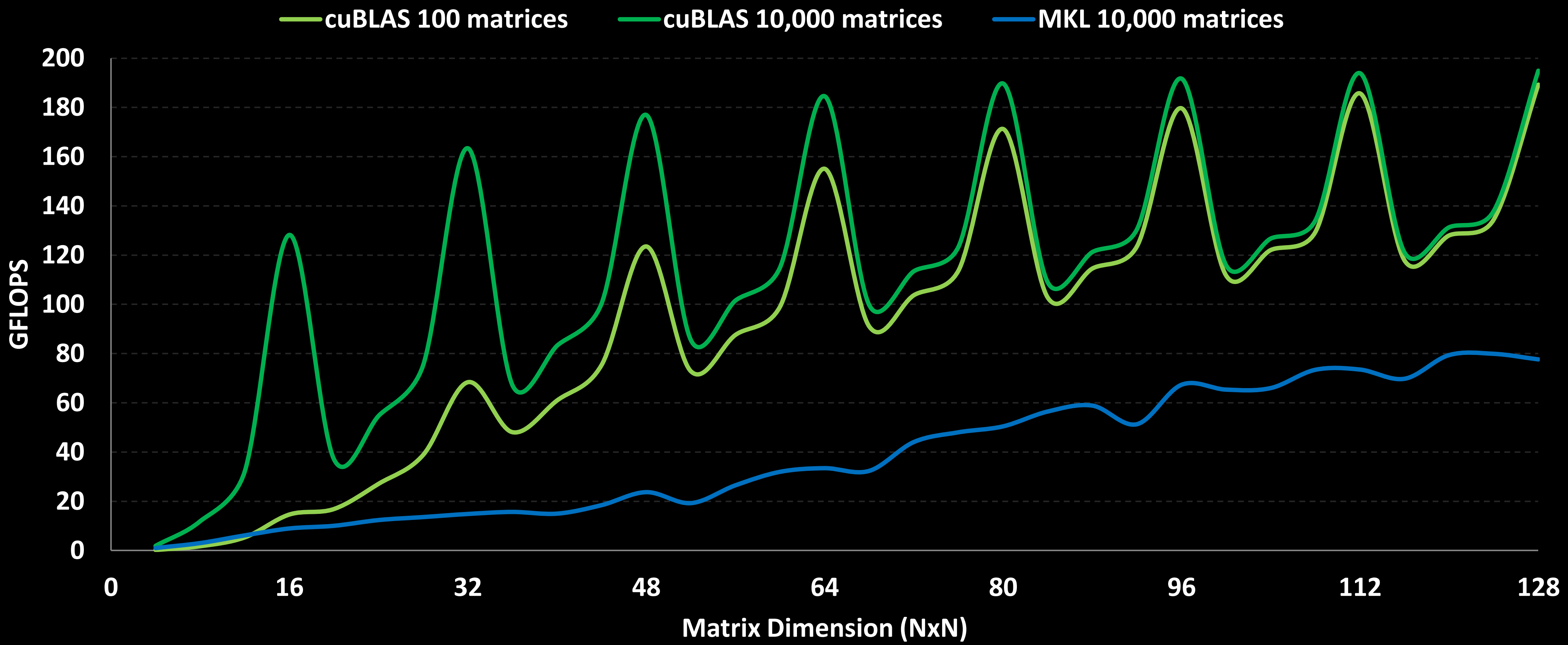
ZGEMM Performance vs Intel MKL



Performance may vary based on OS version and motherboard configuration

- cuBLAS 4.1 on Tesla M2090, ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

cuBLAS Batched GEMM API improves performance on batches of small matrices



Performance may vary based on OS version and motherboard configuration

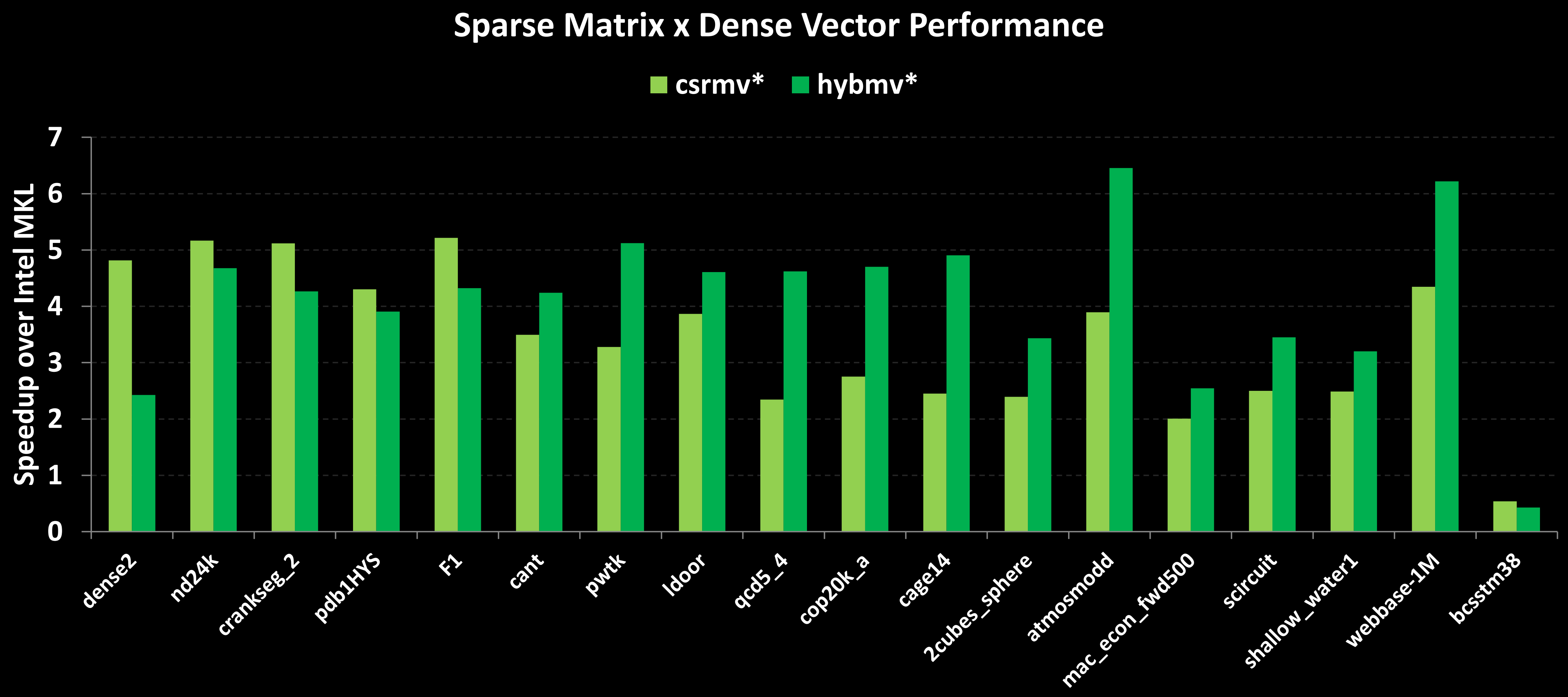
- cuBLAS 4.1 on Tesla M2090, ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

cuSPARSE: Sparse linear algebra routines

- Sparse matrix-vector multiplication & triangular solve
  - APIs optimized for iterative methods
- New in 4.1
  - Tri-diagonal solver with speedups up to 10x over Intel MKL
  - ELL-HYB format offers 2x faster matrix-vector multiplication

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \alpha \begin{bmatrix} 1.0 & & & \\ 2.0 & 3.0 & & \\ & & 4.0 & \\ 5.0 & & 6.0 & 7.0 \end{bmatrix} \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \\ 4.0 \end{bmatrix} + \beta \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

cuSPARSE is >6x Faster than Intel MKL

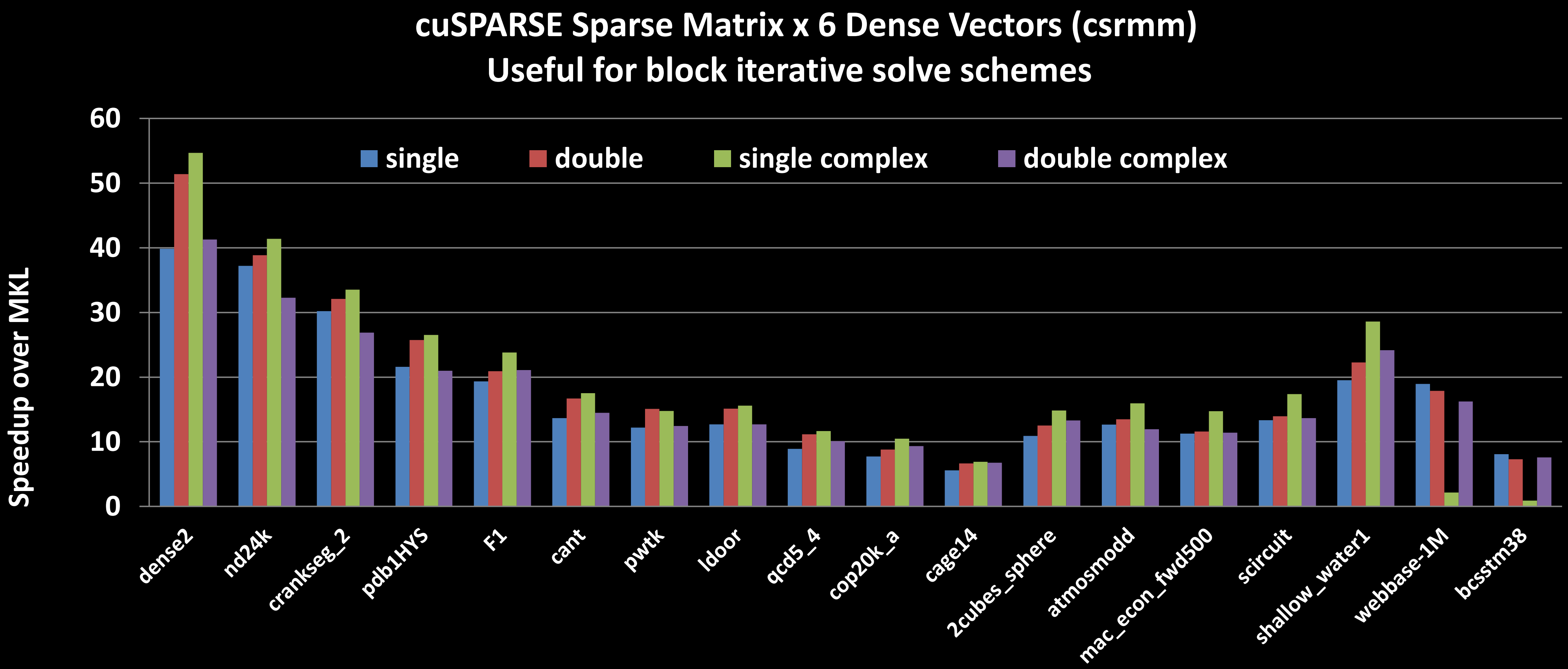


**Average speedup over single, double, single complex & double-complex*

Performance may vary based on OS version and motherboard configuration

- cuSPARSE 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

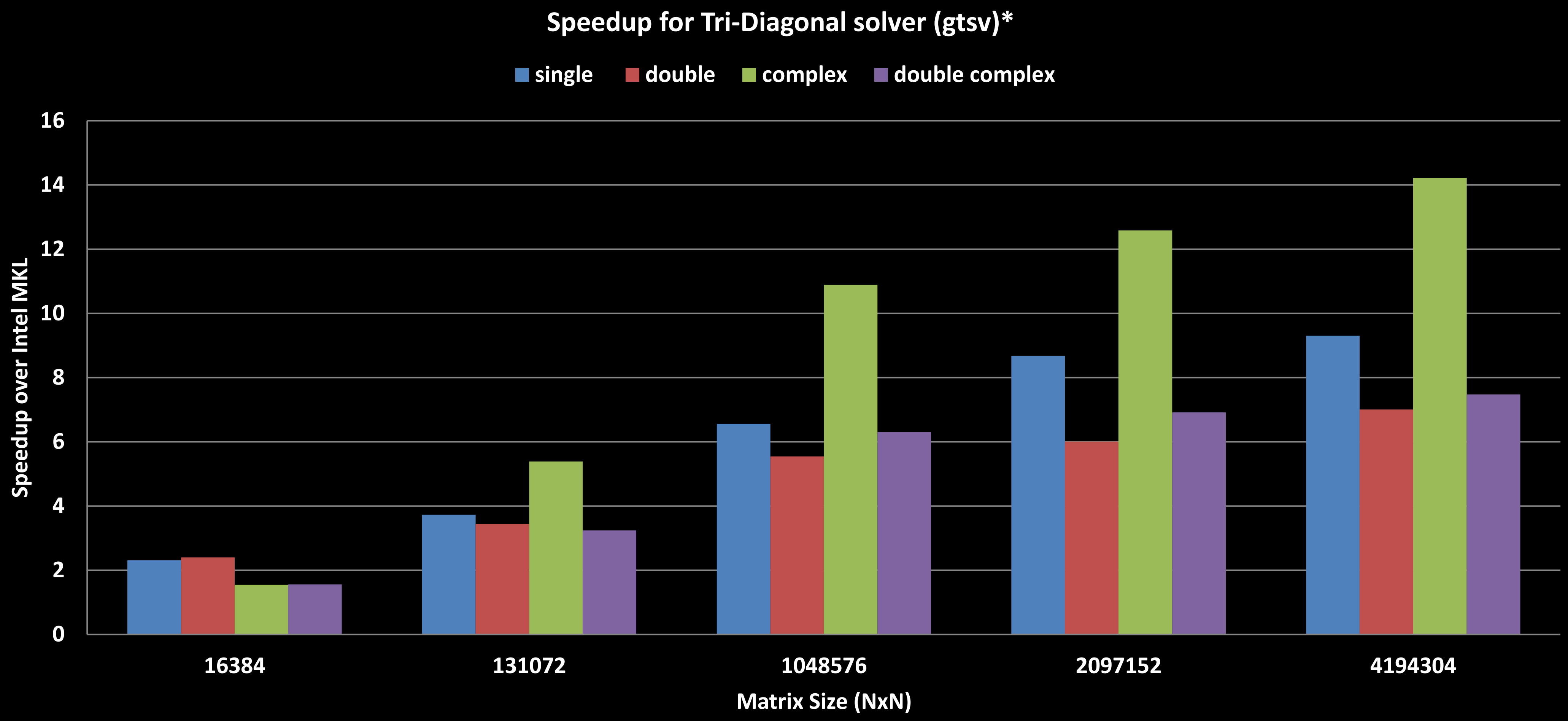
Up to 40x faster with 6 CSR Vectors



Performance may vary based on OS version and motherboard configuration

- cuSPARSE 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Tri-diagonal solver performance vs. MKL



**Parallel GPU implementation does not include pivoting*

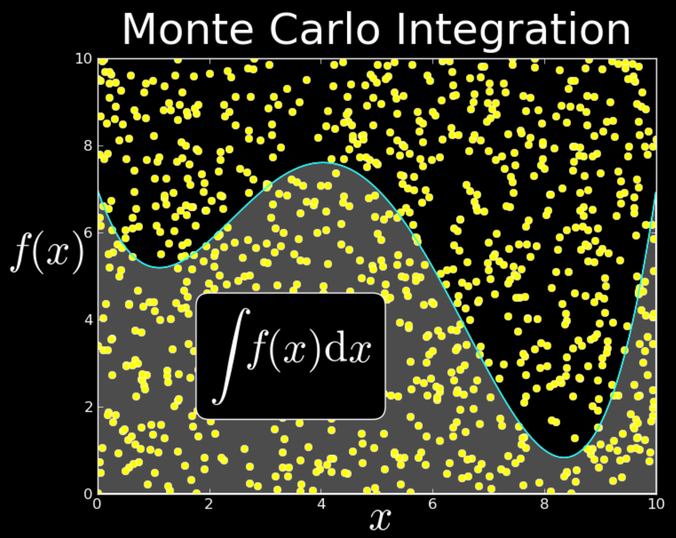
- cuSPARSE 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Performance may vary based on OS version and motherboard configuration

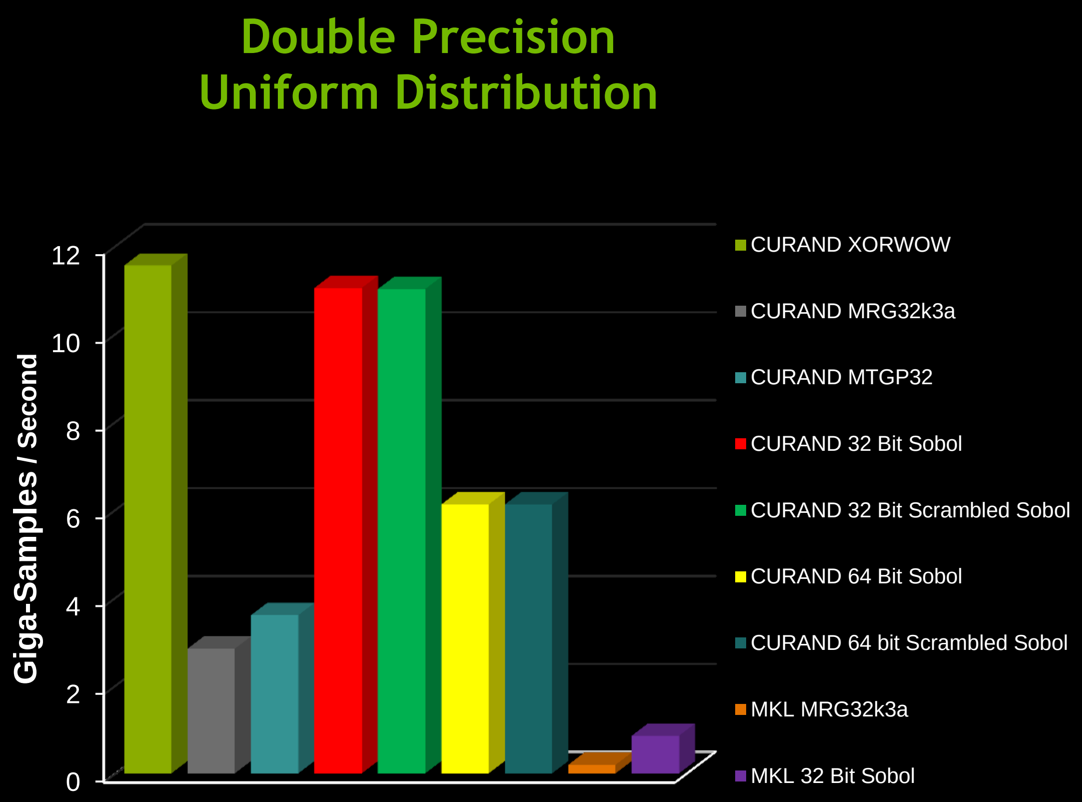
cuRAND: Random Number Generation

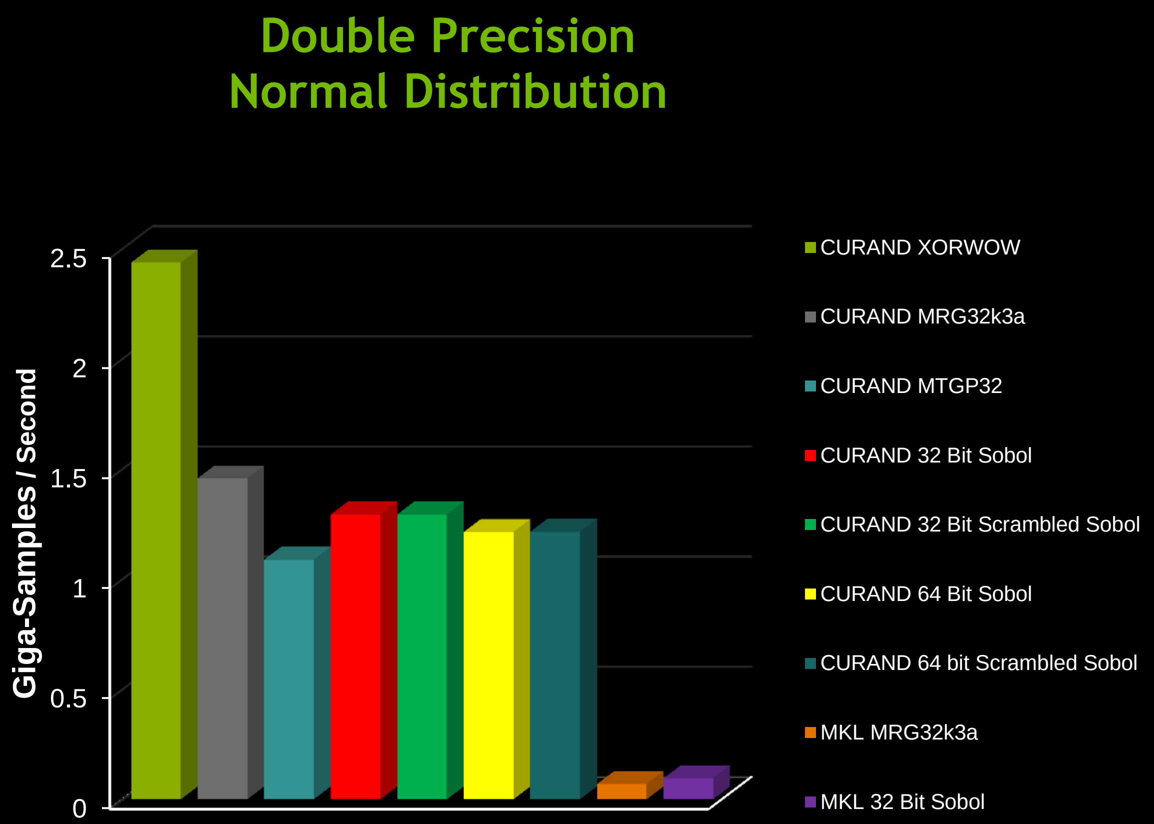
- Pseudo- and Quasi-RNGs
- Supports several output distributions
- Statistical test results reported in documentation

- New commonly used RNGs in CUDA 4.1
  - MRG32k3a RNG
  - MTGP11213 Mersenne Twister RNG



cuRAND Performance compared to Intel MKL





Performance may vary based on OS version and motherboard configuration

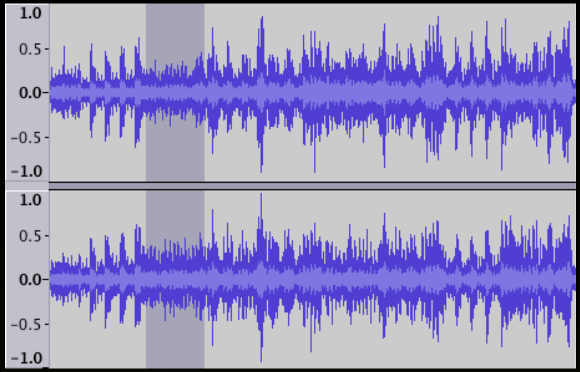
- cuRAND 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 @ 3.33 GHz

1000+ New Imaging Functions in NPP 4.1

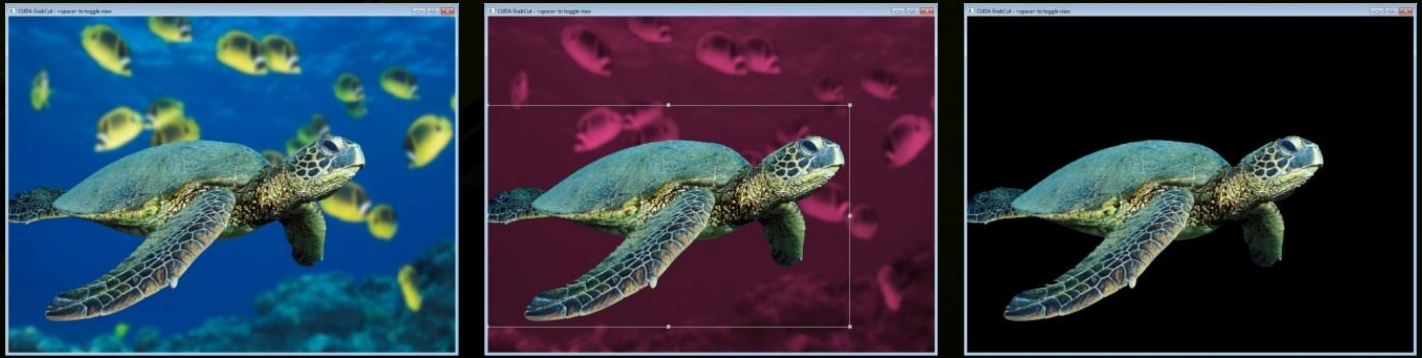
Up to **40x** speedups

- NVIDIA Performance Primitives (NPP) library includes over 2200 GPU-accelerated functions for image & signal processing

  Arithmetic, Logic, Conversions, Filters, Statistics, etc.



- Most are 5x-10x faster than analogous routines in Intel IPP



http://developer.nvidia.com/content/graphcuts-using-npp

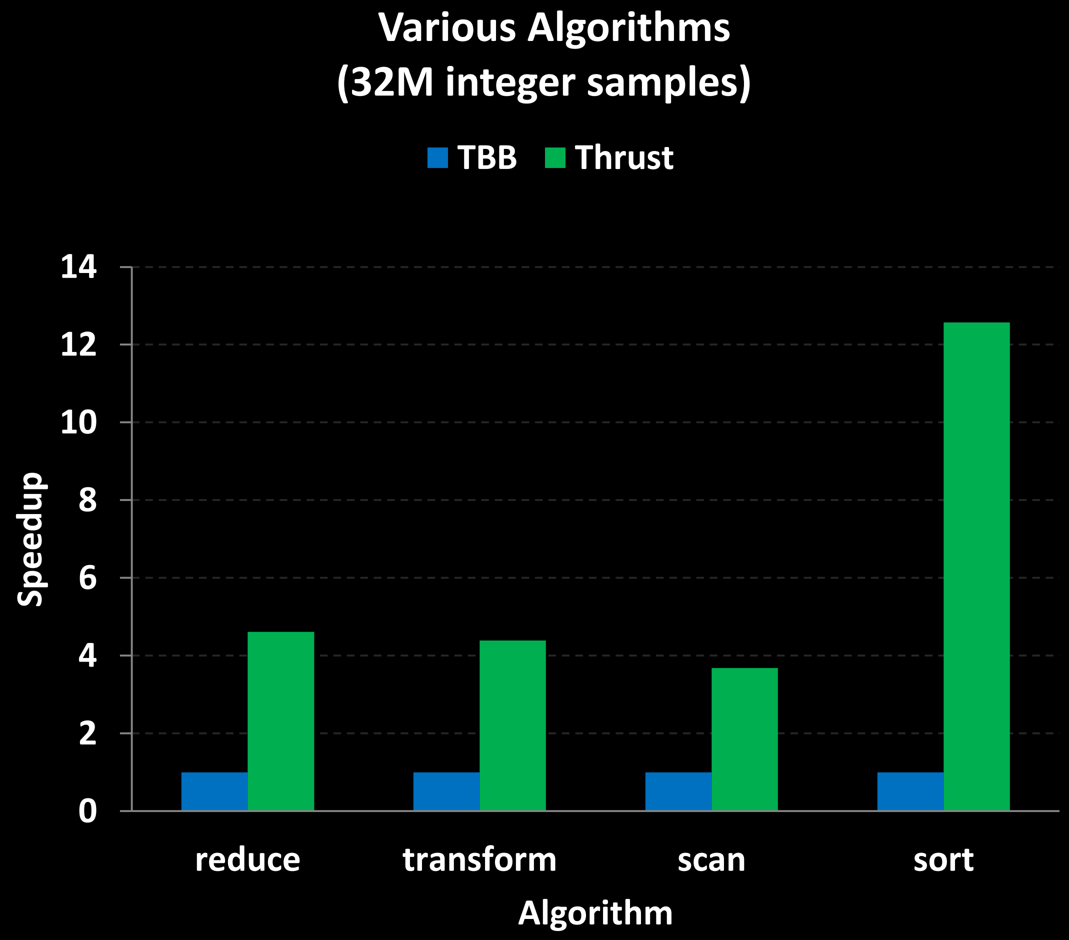
* NPP 4.1, NVIDIA C2050 (Fermi)
* IPP 6.1, Dual Socket Core™ i7 920 @ 2.67GHz

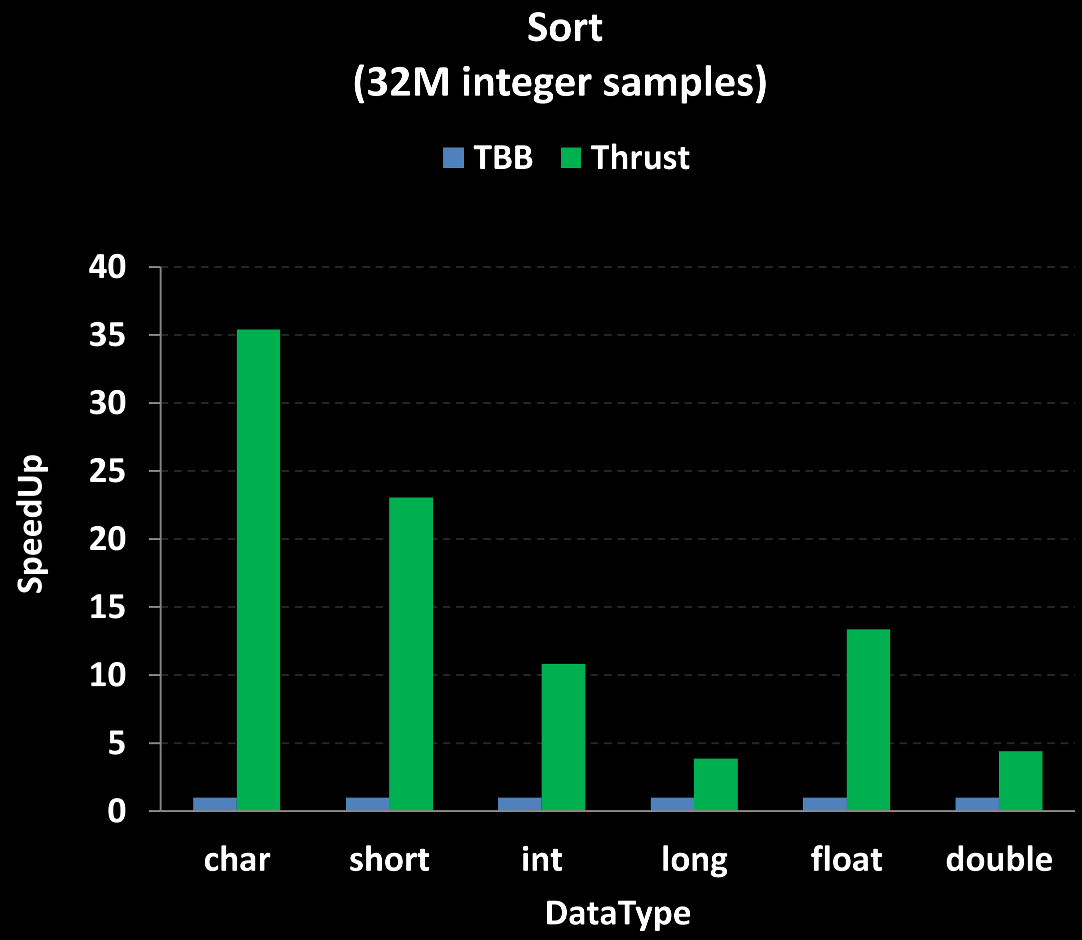
Thrust: CUDA C++ Template Library

- Template library for CUDA mimics the C++ STL
  - Optimized algorithms for sort, reduce, scan, etc.
  - OpenMP backend for portability
- Allows applications and prototypes to be built *quickly*
- New in 4.1: Boost-style placeholders allow inline functors
  - Example: *saxpy* in 1 line:

```
thrust::transform(x.begin(), x.end(), y.begin(), y.begin(), a * _1 + _2);
```

Thrust Performance compared to Intel TBB





- Thrust 4.1, Tesla M2090 (Fermi), ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

Performance may vary based on OS version and motherboard configuration

math.h: C99 floating-point library + extras

- Basic: +, *, /, 1/, sqrt, FMA (all IEEE-754 accurate for float, double, all rounding modes)
- Exponentials: exp, exp2, log, log2, log10, ...
- Trigonometry: sin, cos, tan, asin, acos, atan2, sinh, cosh, asinh, acosh, ...
- Special functions: lgamma, tgamma, erf, erfc
- Utility: fmod, remquo, modf, trunc, round, ceil, floor, fabs, ...
- Extras: rsqrt, rcbrt, exp10, sinpi, sincos, cospi, erfinv, erfcinv, ...

- New in 4.1
  - Bessel functions: j0, j1, jn, y0, y1, yn
  - Scaled complementary error function: erfcx
  - Average and rounded average: __{u}hadd, __{u}rhadd

GPU Technology Conference 2012
May 14-17 | San Jose, CA

The one event you can't afford to miss

- Learn about leading-edge advances in GPU computing
- Explore the research as well as the commercial applications
- Discover advances in computational visualization
- Take a deep dive into parallel programming

Ways to participate

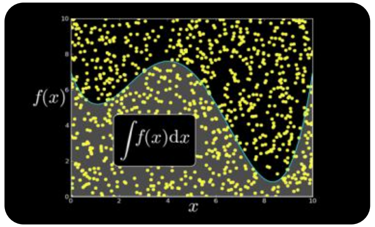
- Speak – share your work and gain exposure as a thought leader
- Register – learn from the experts and network with your peers
- Exhibit/Sponsor – promote your company as a key player in the GPU ecosystem

www.gputechconf.com

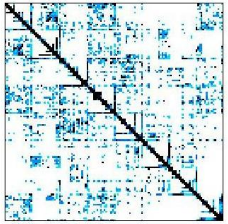
ADDITIONAL SLIDES…

Ecosystem Update

NVIDIA cuBLAS



NVIDIA cuRAND



NVIDIA cuSPARSE

NVIDIA NPP

GPU VSIPL

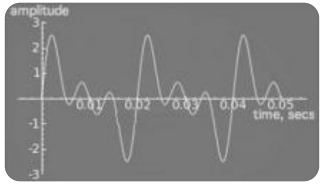
Vector Signal
Image Processing

CULA tools

GPU Accelerated
Linear Algebra

MAGMA

Matrix Algebra on
GPU and Multicore



NVIDIA cuFFT

ROGUE WAVE
SOFTWARE

IMSL Library

CUSP

Sparse Linear
Algebra

libjacket

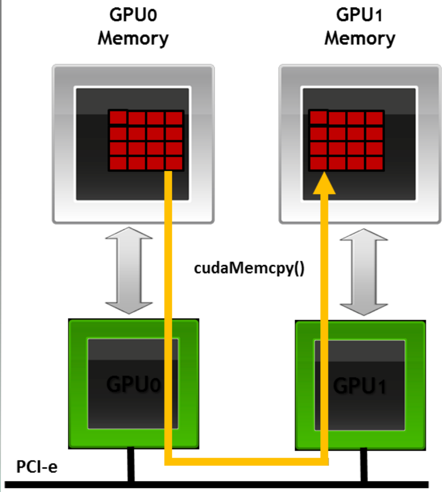
Building-block
Algorithms for CUDA

Thrust

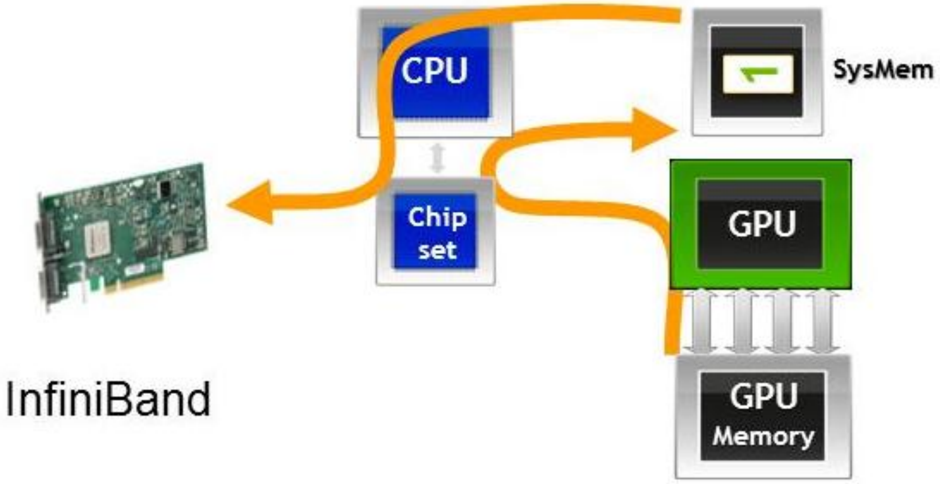
C++ STL Features
for CUDA

GPU Accelerated Libraries

"Drop-in" Acceleration for Your Applications



Peer-to-Peer Transfers



GPU Direct

OPENFABRICS ALLIANCE

As of OFED 1.5.2

MVAPICH

Pre-Release

Announced at SC'11

Platform Computing

Platform MPI

Beta announced at SC'11

GPU-Aware MPI Libraries

Integrated Support for GPU Computing

Programming Languages & APIs

CUDA Fortran

PGI Accelerator

HMPP Compiler

CUDA-x86

OpenCL

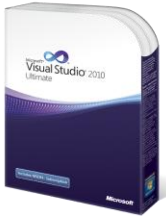
Python for CUDA

NVIDIA CUDA

NVIDIA C Compiler

OpenGL

Microsoft® DirectX® 11



Microsoft AMP C/C++

Debugging Solutions

Command Line to Cluster Wide

NVIDIA Parallel Nsight
for Visual Studio

NVIDIA CUDA-MEMCHECK
for Linux & Mac

Allinea DDT with CUDA
Distributed Debugging Tool

NVIDIA CUDA-GDB
for Linux & Mac

TotalView for CUDA
for Linux Clusters

Performance Analysis Tools

Single GPU to Hybrid Cluster Solutions

NVIDIA Parallel Nsight
for Visual Studio

Vampir Trace Collector

NVIDIA Visual Profiler
for Linux & Mac

Tuning and Analysis Utilities

TAU Performance System

PAPI

PAPI CUDA Component

HPCToolkit

Under Development

Platform Computing

LSF, HPC, Cluster Manager

Bright Computing

Bright Cluster Manager

Adaptive Computing

ROCKS+ MOAB

PBS Works™

PBS Professional

Ganglia

NVML Plugin for GPUs

UNIVA

Univa Grid Engine

Job Scheduling & Cluster Management

GPU Technology Conference Worldwide Events

GTC Asia, Beijing, December 14-15, 2011

Focusing on the very latest scientific research and commercial applications in GPU computing.

GTC 2012, San Jose, CA, May 14-17, 2012

Advancing awareness of High Performance Computing and the transformational impact of GPUs.

www.gputechconf.com

GPU Technology Conference 2012
May 14-17 | San Jose, CA

The one event you can't afford to miss

- Learn about leading-edge advances in GPU computing
- Explore the research as well as the commercial applications
- Discover advances in computational visualization
- Take a deep dive into parallel programming

Ways to participate

- Speak – share your work and gain exposure as a thought leader
- Register – learn from the experts and network with your peers
- Exhibit/Sponsor – promote your company as a key player in the GPU ecosystem

www.gputechconf.com

Co-located with GTC 2012…

Accelerated High Performance Computing Symposium (AHPC)

Hosted by Los Alamos National Laboratory & NVIDIA

- Learn how accelerator technologies can be leveraged in innovative ways to advance the state-of-the-art for simulations on large-scale systems
- Identify hardware and software requirements that can meet the requirements of power, scalability and fault tolerance needed for the next generation of HPC
- Understand how legacy codes can be adapted to make use of modern computing architectures
- Provide feedback to the vendor community to aid in the adoption of accelerator technologies

> "The growing success of GTC makes it a natural venue for co-hosting the Accelerated HPC Symposium. This event draws senior scientists from national research labs across the globe, and their interests in hardware and software development make for a perfect match with GTC."
>
> ~Ben Bergen, Research Scientist, Los Alamos National Laboratory

Sign up for announcements at www.gputechconf.com

CUDA 4.1

The ‘Super’ Computing Company

From Super Phones to Super Computers