

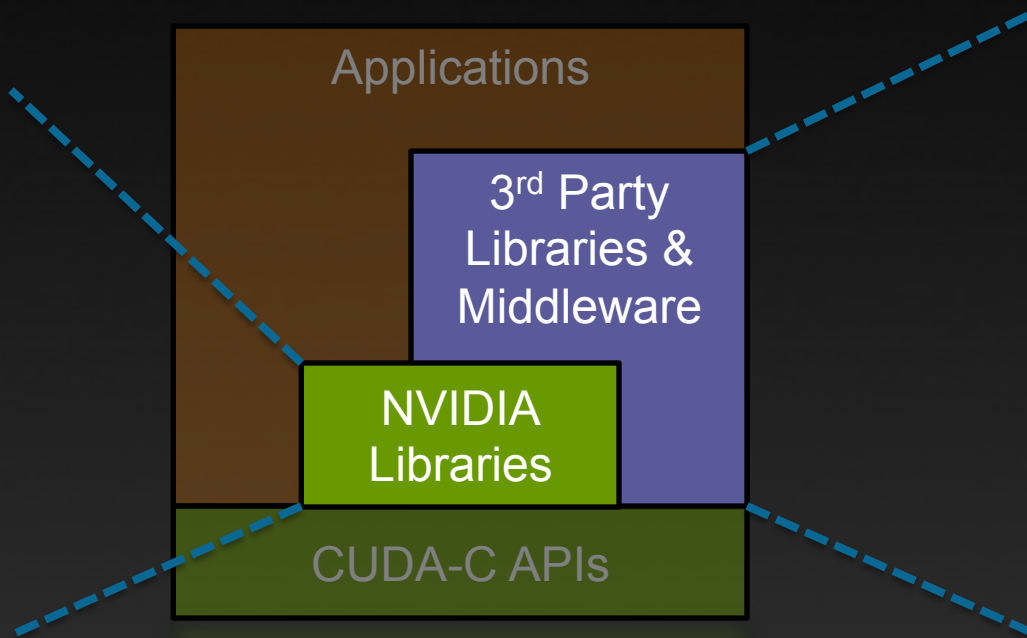
NVIDIA CUDA Libraries

Ujval Kapasi
March 29, 2012



CUDA Libraries and Middleware

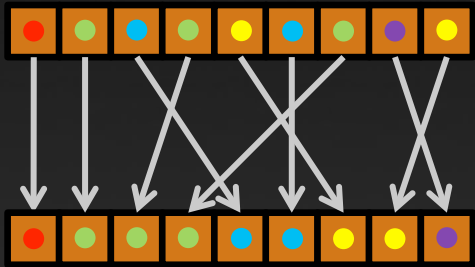
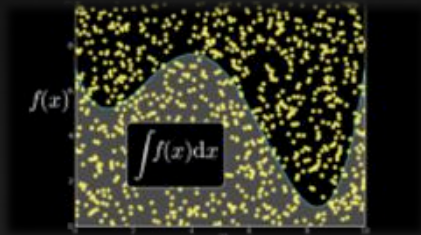
Basic building blocks
Easy to use
Fast



Programming language integration
Hybrid computation
Multi-node and multi-gpu
Domain specific languages



NVIDIA CUDA Libraries



CUBLAS

dense linear algebra

CUSPARSE

sparse linear algebra

CUFFT

discrete Fourier transforms

CURAND

random number generation

NPP

signal and image processing

Thrust

scan, sort, reduce, transform

math.h

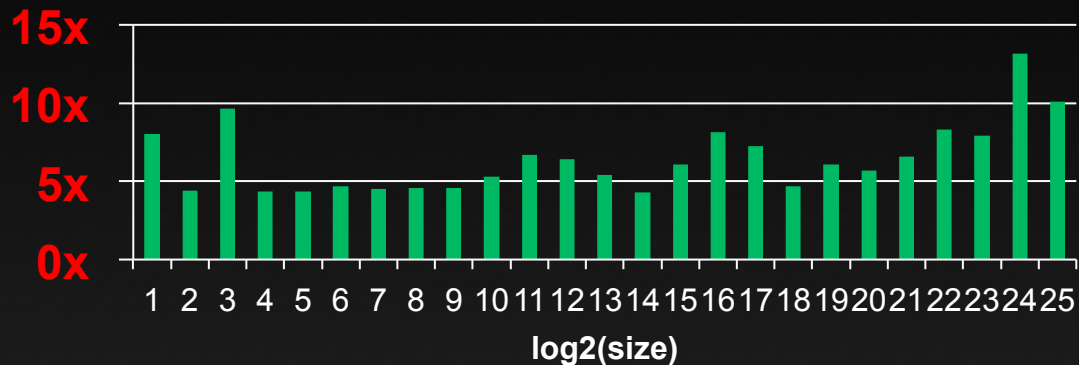
floating point

system calls

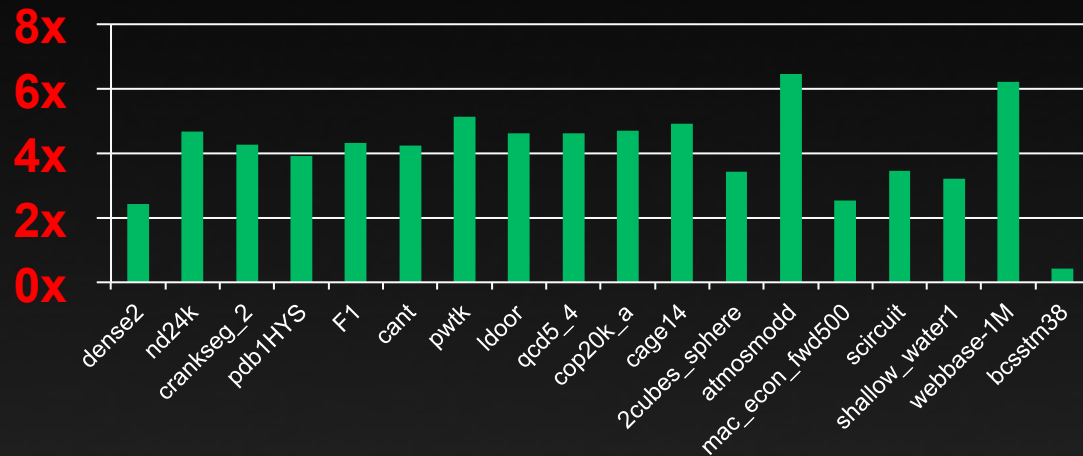
printf, malloc, assert

4x-10x speedups over CPU on single precision

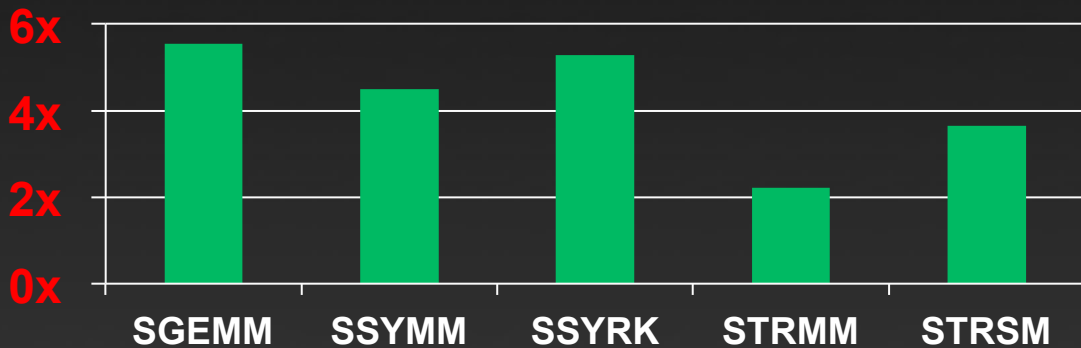
CUFFT



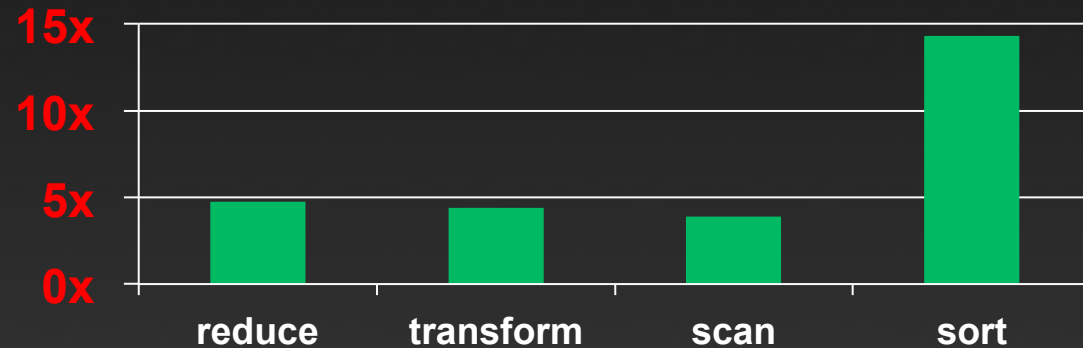
CUSPARSE



CUBLAS



Thrust



NOTE: ROUTINES ASSUME INPUT AND OUTPUT DATA ARE IN GPU MEMORY

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

Challenge #1: MATCH AN ESTABLISHED CPU ECOSYSTEM

2007

CUDA Toolkit
1.x

- Single precision
- cuBLAS
- cuFFT
- math.h

2008

CUDA Toolkit
2.x

- Double Precision support in all libraries

2009

2010

CUDA Toolkit
3.x

- cuSPARSE
- cuRAND
- printf()
- malloc()

2011

CUDA Toolkit
4.x

- Thrust
- NPP
- assert()

CUDA ecosystem today spans many domains

- Languages (FORTRAN, python)
- Math, Numerics, Statistics
- Dense & Sparse Linear Algebra
- Algorithms (sort, etc.)
- Image Processing
- Computer Vision
- Signal Processing
- Finance

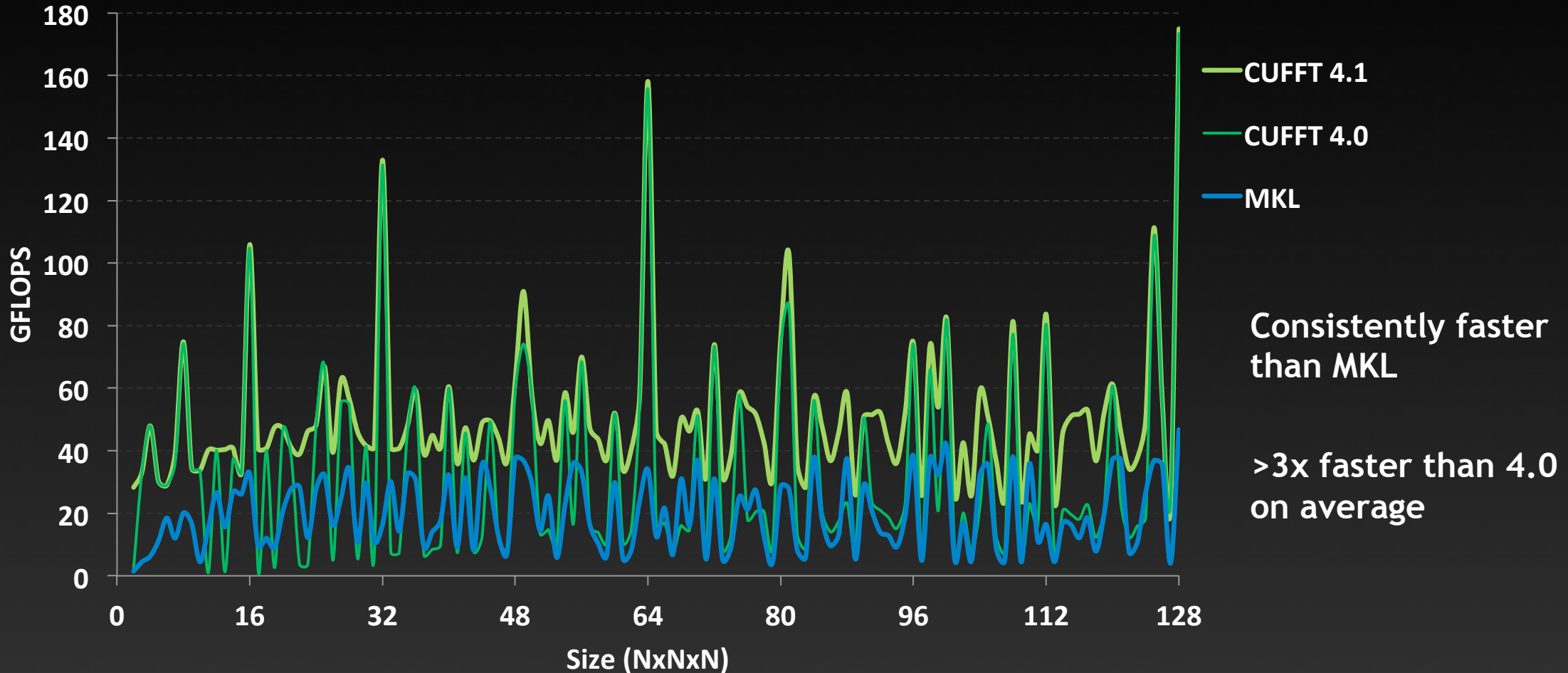
GPU-Accelerated Libraries

Adding GPU-acceleration to your application can be as easy as simply calling a library function. Check out the extensive list of high performance GPU-accelerated libraries below. If you would like other libraries added to this list please [contact us](#).

 <p>NVIDIA cuFFT NVIDIA CUDA Fast Fourier Transform Library (cuFFT) provides a simple interface for computing FFTs up to 10x faster, without having to develop your own custom GPU FFT implementation.</p>	 <p>NVIDIA cuBLAS NVIDIA CUDA BLAS Library (cuBLAS) is a GPU-accelerated version of the complete standard BLAS library that delivers 6x to 17x faster performance than the latest MKL BLAS.</p>	 <p>CULA Tools GPU-accelerated linear algebra library by EM Photonics, that utilizes CUDA to dramatically improve the computation speed of sophisticated mathematics.</p>
 <p>MAGMA A collection of next gen linear algebra routines. Designed for heterogeneous GPU-based architectures. Supports current LAPACK and BLAS standards.</p>	 <p>IMSL Fortran Numerical Library Developed by RogueWave, a comprehensive set of mathematical and statistical functions that offloads work to GPUs.</p>	 <p>NVIDIA cuSPARSE NVIDIA CUDA Sparse (cuSPARSE) Matrix library provides a collection of basic linear algebra subroutines used for sparse matrices that delivers over 8x performance boost.</p>
 <p>NVIDIA CUSP An GPU accelerated Open Source C++ library of generic parallel algorithms for sparse linear algebra and graph computations. Provides a easy to use high-level interface.</p>	 <p>AccelerEyes LibJacket Comprehensive GPU function library, including functions for math, signal and image processing, statistics, and more. Interfaces for C, C++, Fortran, and Python.</p>	 <p>NVIDIA cuRAND The CUDA Random Number Generation library performs high quality GPU-accelerated random number generation (RNG) over 8x faster than typical CPU only code.</p>
 <p>NVIDIA NPP</p>	 <p>NVIDIA CUDA Math library</p>	 <p>Thrust</p>

Challenge #2: PREDICTABLE ACCELERATION

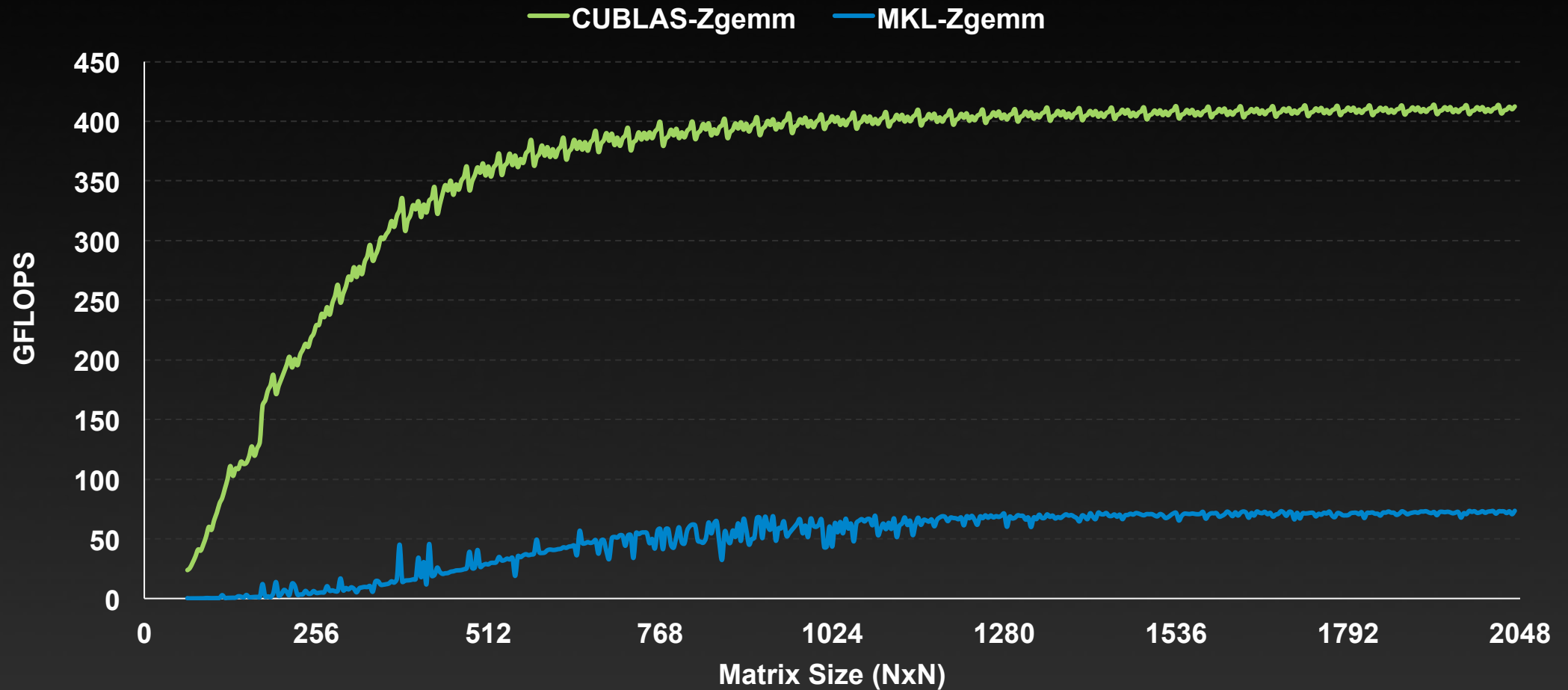
Single Precision All Sizes 2x2x2 to 128x128x128



NOTE: ROUTINES ASSUME INPUT AND OUTPUT DATA ARE IN GPU MEMORY
Performance may vary based on OS version and motherboard configuration

- cuFFT 4.1 and cuFFT 4.0 on Tesla M2090, ECC on
- MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz

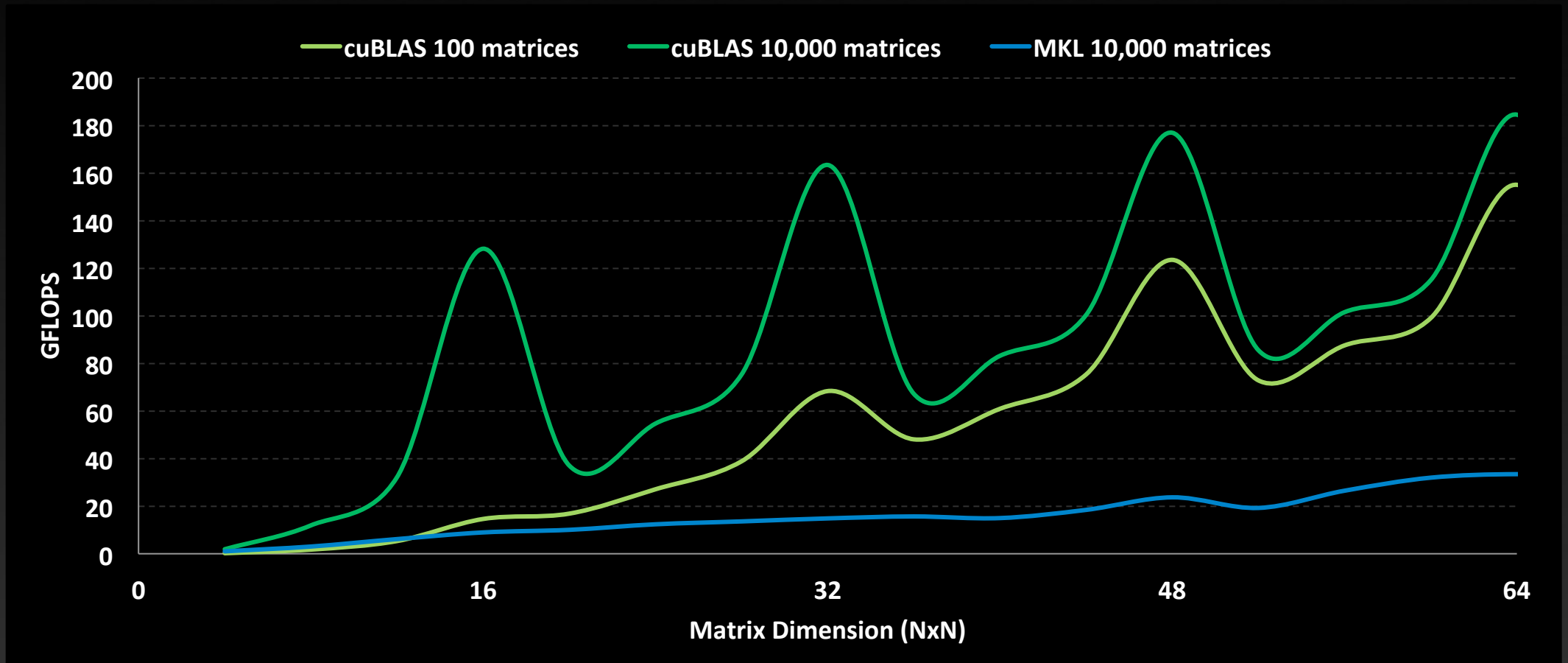
Challenge #2: PREDICTABLE ACCELERATION



NOTE: ROUTINES ASSUME INPUT AND OUTPUT DATA ARE IN GPU MEMORY
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

Challenge #2: PREDICTABLE ACCELERATION



NOTE: ROUTINES ASSUME INPUT AND OUTPUT DATA ARE IN GPU MEMORY
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

- **CUDA Libraries accelerate basic building block algorithms**
 - BLAS → CUBLAS, FFTW → CUFFT, STL → Thrust
 - Enables a wide range of 3rd party libraries and middleware
- **CUDA Libraries coverage and performance constantly improving**
 - Automatic performance improvements with new CUDA releases and new GPU architectures

Thank You!

