NVIDIA GTC

CONNECT WITH THE EXPERTS

NVIDIA Math Libraries

(CWE 21216)

Harun Bayraktar, Senior Manager, CUDA Math Libraries

Alexander Kalinkin, Manager, cuSOLVER and cuSPARSE

Azzam Haidar, Senior Engineer, cuSOLVER

Samuel Rodriguez Bernabeu, Senior Engineer, cuSOLVER

Piotr Majcher, Senior Engineer, cuBLAS

Markus Hoehnerbach, Senior Engineer, cuTENSOR

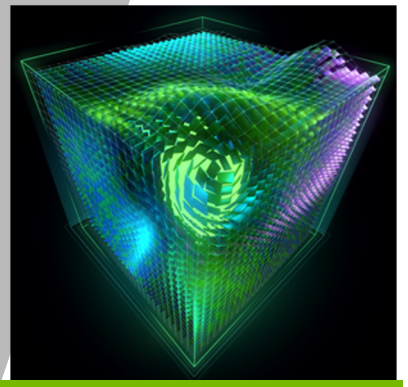
Lukasz Ligowski, Senior Engineer, cuFFT

Mahesh Khadatare, Senior Engineer, NPP and nvJPEG

Zoheb Khan, Senior Engineer, nvJPEG

NVIDIA GTC

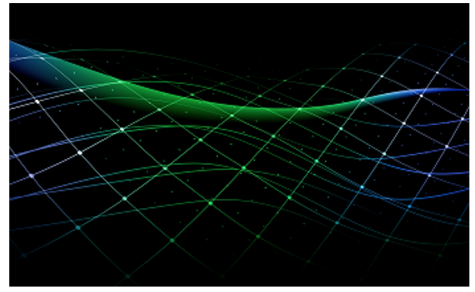
CUDA Math Libraries



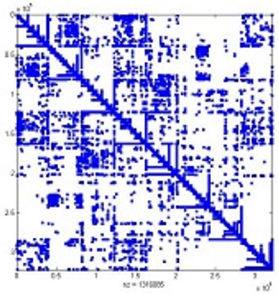
CUDA Math API

cuBLAS

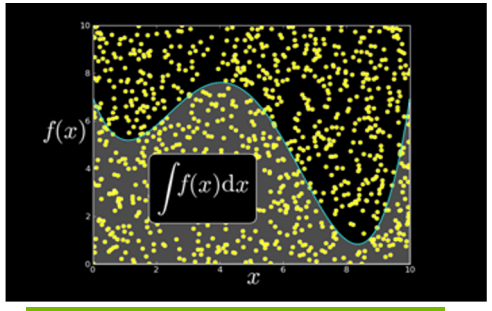
cuTENSOR



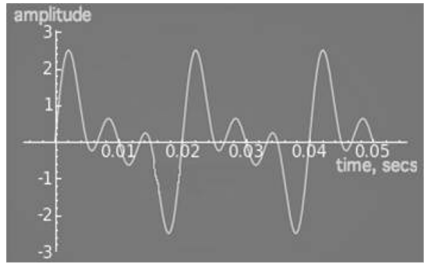
cuSOLVER



cuSPARSE



cuRAND



cuFFT

NPP

nvJPEG

NVIDIA GTC

Key Developments

cuSOLVER added
- multi-GPU eigensolver and linear solver capabilities
- mixed-precision tensor core accelerated linear solver

cuBLAS Tensor Core support on Volta and Turing architecture GPUs is extended and performance is continually improved

cuFFTDx (cuFFT Device Extension) library provides ability to inline NVIDIA optimized kernels to fuse with other operations
developer.nvidia.com/CUDAMathLibraryEA

nvJPEG added encoding functionality and extended decoder functionality

16 x GV100 GPU DGX-2

NVIDIA GTC

CONNECT WITH THE EXPERTS

NVIDIA Math Libraries

(CWE 21216)

Harun Bayraktar, Senior Manager, CUDA Math Libraries

Alexander Kalinkin, Manager, cuSOLVER and cuSPARSE

Azzam Haidar, Senior Engineer, cuSOLVER

Samuel Rodriguez Bernabeu, Senior Engineer, cuSOLVER

Piotr Majcher, Senior Engineer, cuBLAS

Markus Hoehnerbach, Senior Engineer, cuTENSOR

Lukasz Ligowski, Senior Engineer, cuFFT

Mahesh Khadatare, Senior Engineer, NPP and nvJPEG

Zoheb Khan, Senior Engineer, nvJPEG

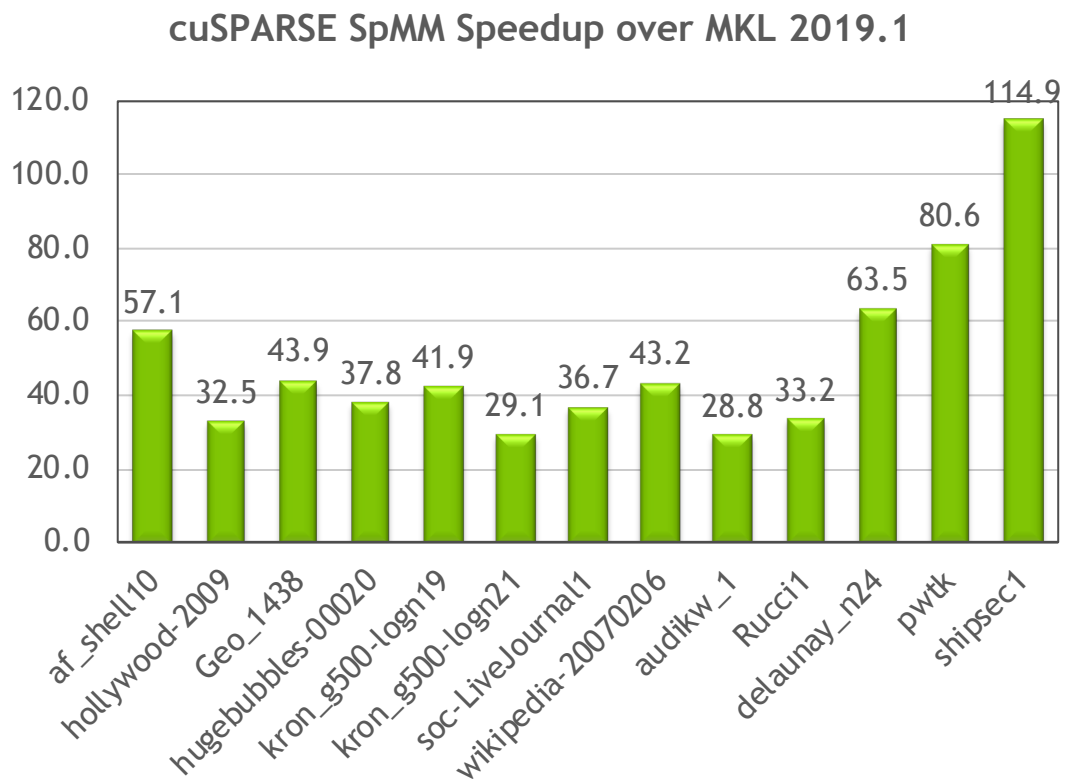
BACKUP MATERIAL

cuSPARSE

Introduced generic APIs with improved performance

- SpVV - Sparse Vector Dense Vector Multiplication
- SpMV – Sparse Matrix Dense Vector Multiplication
- SpMM – Sparse Matrix Dense Matrix Multiplication

```
cusparseStatus_t
cusparseSpMM(cusparseHandle_t              handle,
             cusparseOperation_t           transA,
             cusparseOperation_t           transB,
             const void*                   alpha,
             const cusparseSpMatDescr_t    matA,
             const cusparseDenseMatDescr_t matB,
             const void*                   beta,
             cusparseDenseMatDescr_t       matC,
             cudaDataType                  computeType,
             cusparseSpMMAlg_t             alg,
             void*                         externalBuffer)
```



cuBLAS 10.1 Update 1 performance collected on GV100; MKL 2019.1 performance collected on 2-socket Xeon Gold 6140

cuBLAS Highlights

CUDA 10.1 introduced new cuBLASLt

Advanced matmul functionality

CUDA 10.1 Update 2

Relaxation of problem size limitation for Tensor Core acceleration

CUDA 10.2

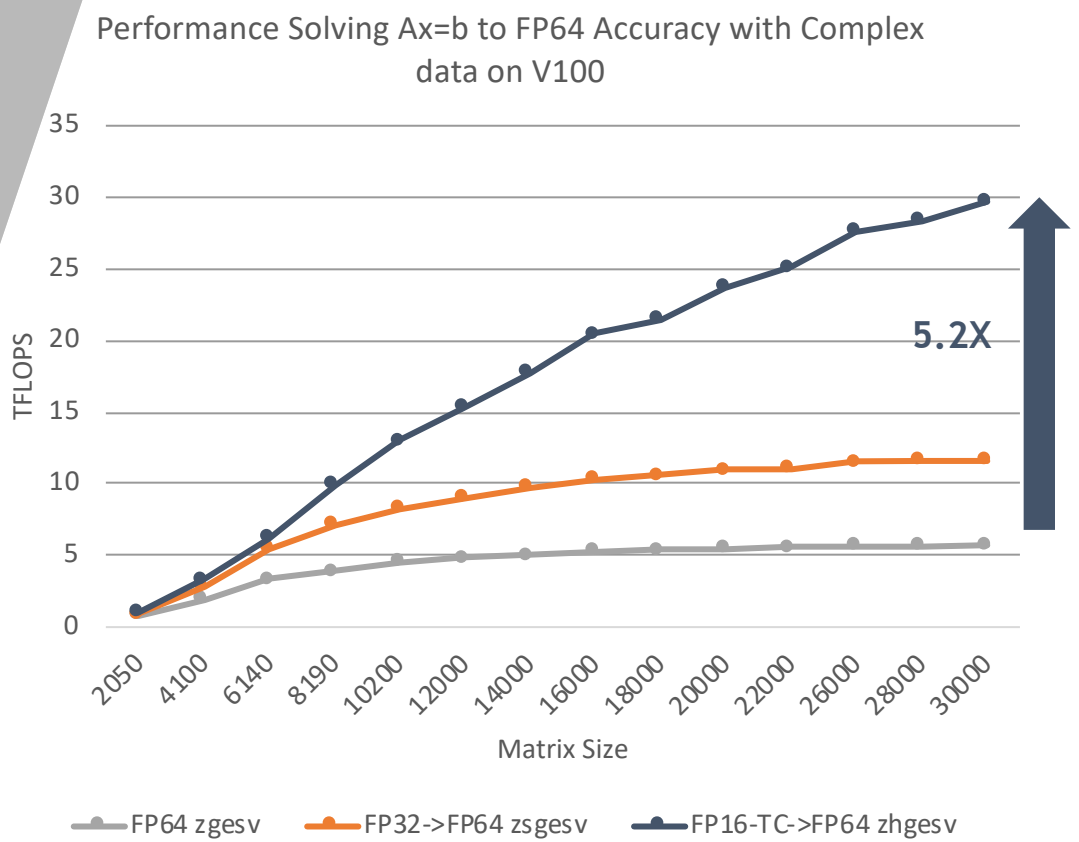
Improved performance on some large and other GEMM sizes due to increased internal workspace size

cuBLASMg EA Release

Single-Process Multi-GPU GEMM with best-in-class, asymptotically peak performance

Compute Type	Scale Type	A/B Type	C Type
CUDA_R_16F	CUDA_R_16F	CUDA_R_16F	CUDA_R_16F
CUDA_R_32I	CUDA_R_32I	CUDA_R_8I	CUDA_R_32I
	CUDA_R_32F	CUDA_R_8I	CUDA_R_32I
			CUDA_R_8I
CUDA_R_32F	CUDA_R_32F	CUDA_R_16F	CUDA_R_16F
		CUDA_R_8I	CUDA_R_32F
		CUDA_R_16F	CUDA_R_32F
		CUDA_R_32F	CUDA_R_32F
CUDA_R_64F	CUDA_R_64F	CUDA_R_64F	CUDA_R_64F
CUDA_C_32F	CUDA_C_32F	CUDA_C_8I	CUDA_C_32F
		CUDA_C_16F	CUDA_C_32F
		CUDA_C_16F	CUDA_C_16F
		CUDA_C_32F	CUDA_C_32F
CUDA_C_64F	CUDA_C_64F	CUDA_C_64F	CUDA_C_64F

cuSOLVER: TENSOR CORE ACCELERATED LINEAR SOLVERS



LU Solver

- CUDA Toolkit 10.2
- Real & Complex FP32 & FP64

Solve dense linear system by one-sided factorizations
Supports Real and Complex, FP32 and FP64 data
Supports multiple right-hand sides
Retain FP64 accuracy with up to 5X Speedup

MATH LIBRARY DEVICE EXTENSIONS

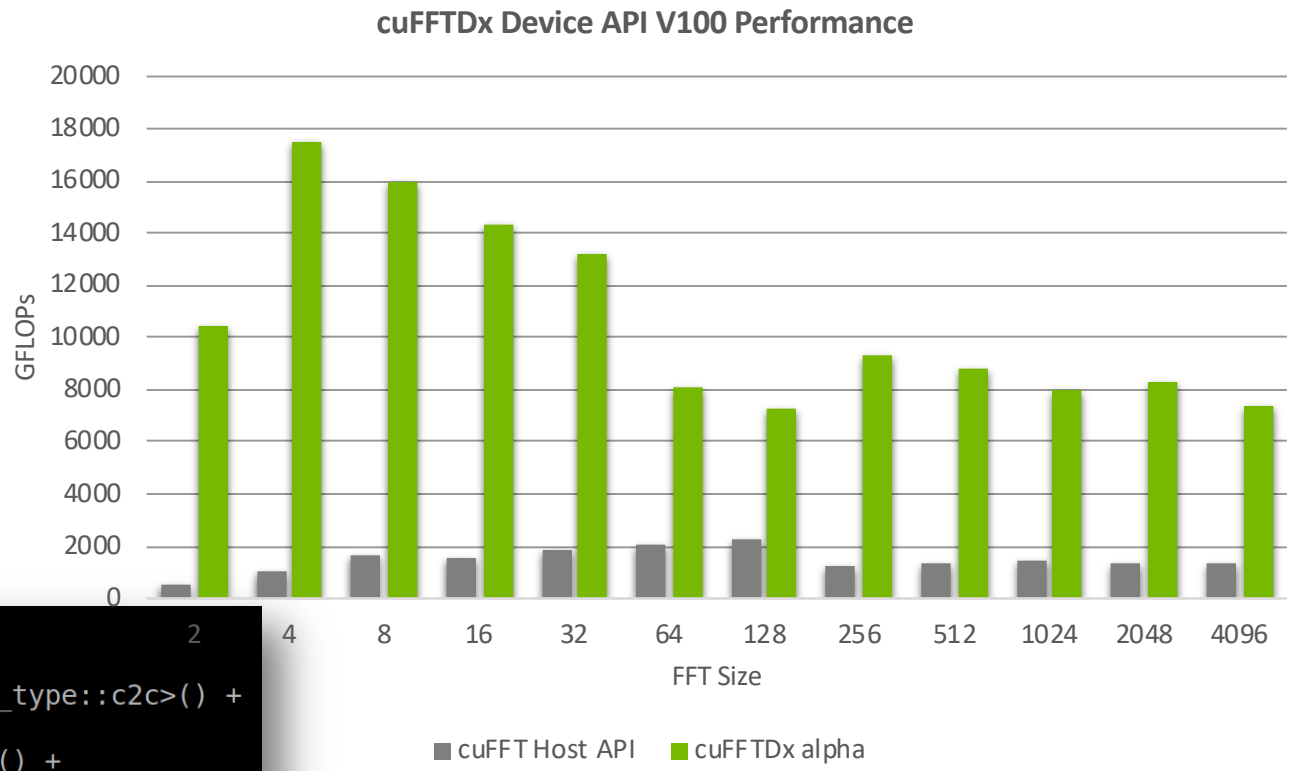
cuFFTDx available in Math Library EA Program

Device callable library

Retain and reuse on-chip data

Inline FFTs in user kernel

Combine FFT operations




```
using namespace cufftdx;

using FFT = decltype(Block() + Size<128>() + Type<fft_type::c2c>() +
            Direction<fft_direction::forward>() +
            Precision<float>() + ElementsPerThread<8>() +
            FFTsPerBlock<2>() + SM<700>());
```

developer.nvidia.com/CUDAMathLibraryEA

cuTENSOR

A New High Performance CUDA Library for Tensor Primitives

v1.0.1 available now at developer.nvidia.com/cutensor

Tensor Contractions and Reductions

Elementwise Operations

Mixed Precision Support

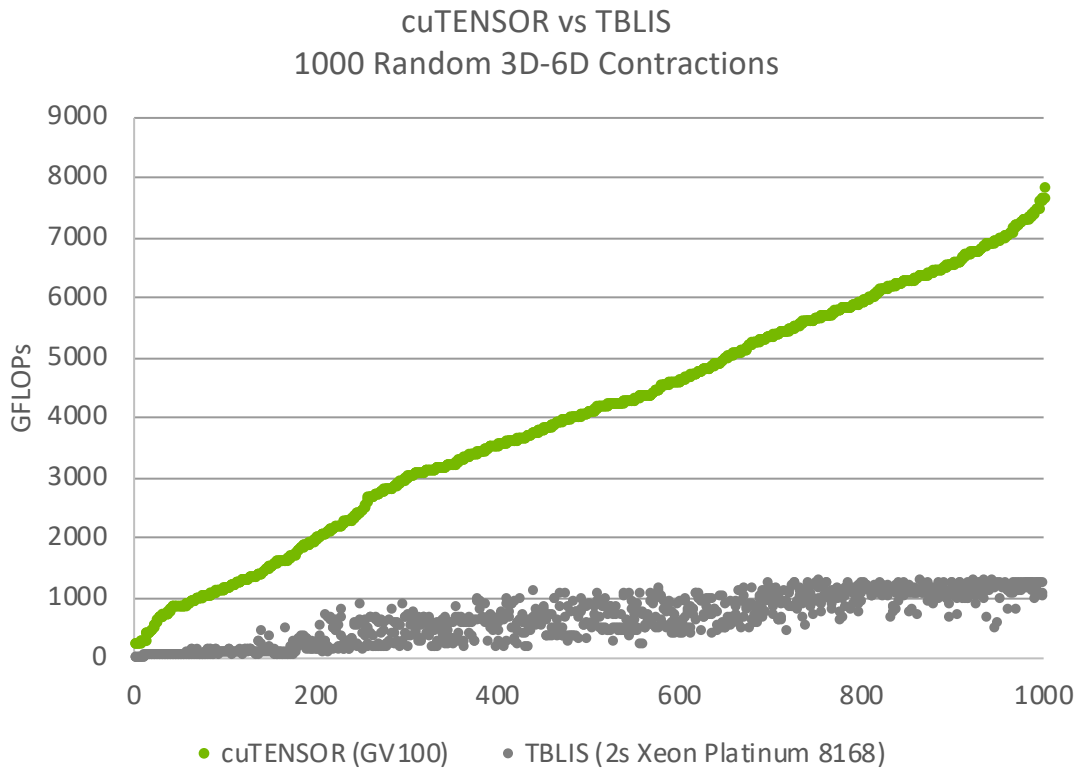
Elementwise Fusion

Tensor Core Acceleration

Impact

Deep learning frameworks aggressively adopting elementwise operations

Up to 23X application end-to-end speedup over previously CPU-only Quantum Chemistry simulations from drop-in contraction API



NVIDIA GTC

nvJPEG

Features in CUDA 10.2 Toolkit

Encoding Feature	Supported Options
Encoding grayscale image	Yes
Interleaved scans support	Yes
Scans type	Baseline, Progressive
Entropy compression	Huffman (optimized and non-optimized)
Restart markers support	No
Quality control	8-bit quantization tables based on 1-100 range quality argument
Target JPEG chroma subsampling	4:4:4, 4:2:2, 4:2:0, 4:4:0, 4:1:0, 4:1:1
JPEG image bits per sample	8
Input data color space	RGB, YUV
Input image format	RGB/BGR interleaved and non-interleaved, Planar YUV in supported chroma subsampling

Decoding Feature	Supported Options
Supported color components number	1 (grayscale), 3, upto 4 channels
Interleaved scans	Yes
Non-interleaved scans	Yes
Huffman entropy decode	Yes (Baseline + progressive)
Arithmetic entropy decode	No
Restart markers support	Yes
Quantization tables precision	8- and 16-bit
Chroma subsampling color conversion	4:4:4, 4:2:2, 4:2:0, 4:4:0, 4:1:0, 4:1:1
Bits per sample	8
Output format	RGB, BGR, RGB interleaved, BGR interleaved, Grayscale. Raw image data without color conversion.