
Rafi (remote)
NVIDIA
TOOLS & LIBRARIES

- NSIGHT Developer tools
- Current offerings
- Analysis – debugging, memory: memcheck, sanitizer
- Profiling – nvprof, CUPTI, visual profiling

CUDA-MEMCHECK

Support for host API functions with pitch parameter.

Support for the Cooperative Groups programming model.

Support for shared memory atomic instructions.

Support for detecting invalid accesses to global memory on Pascal and later architectures that extend beyond the end of an allocation.

Support for limiting the numbers of errors printed by cuda-memcheck.

Racecheck analysis reports are assigned a severity level.

Default print level changed from INFO to WARN.

A new command line option to report deprecated instructions even when they are used in safe execution paths.

UPCOMING

“sanitizer”

Lower overhead

Higher independence from the driver

New “sanitizer” API

- Callback API – CUDA events such as memory/kernel allocations
- Patching API – inserts patches for specific memory instructions


PROFILER TOOLS

UPDATES

NVPROF

Many New Metrics:
- Tensor Core Metrics
- L2 Metrics
- Memory Instructions Per Load/Store

Display PCIe Topology

Collect Trace and Profile in same pass (`--trace`)

CUPTI

New Activity Kind: PCIE

New Attribute: Profiling Scope (Device-Level, Context-Level)

Exposes New Metrics

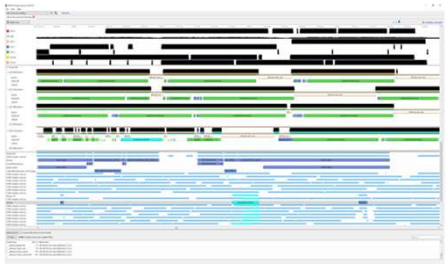
VISUAL PROFILER

Summary View for Memory Hierarchy

Improved Handling of Segments for UVM Data on the Timeline

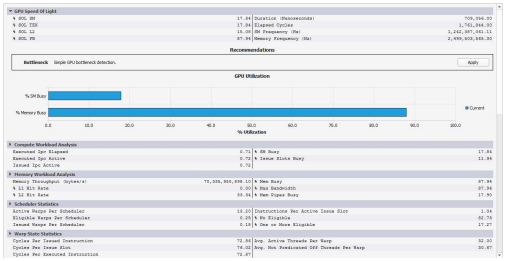

NSIGHT
DEVELOPER TOOLS

NSIGHT PRODUCT FAMILY



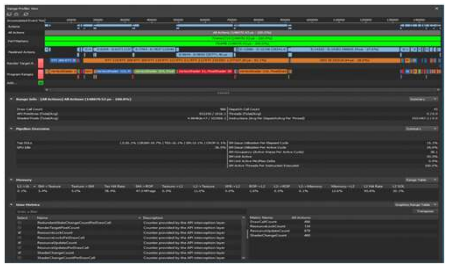
Nsight Systems

System-wide application algorithm tuning



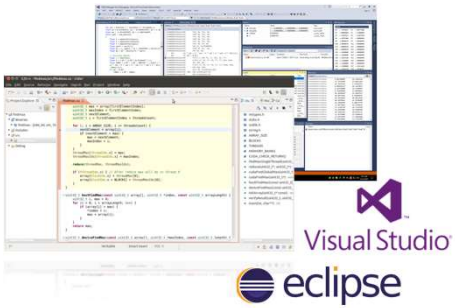
Nsight Compute

CUDA Kernel Profiling and Debugging



Nsight Graphics

Graphics Shader Profiling and Debugging



IDE Plugins

Nsight Eclipse Edition/Visual Studio (Editor, Debugger)

NSIGHT SYSTEMS

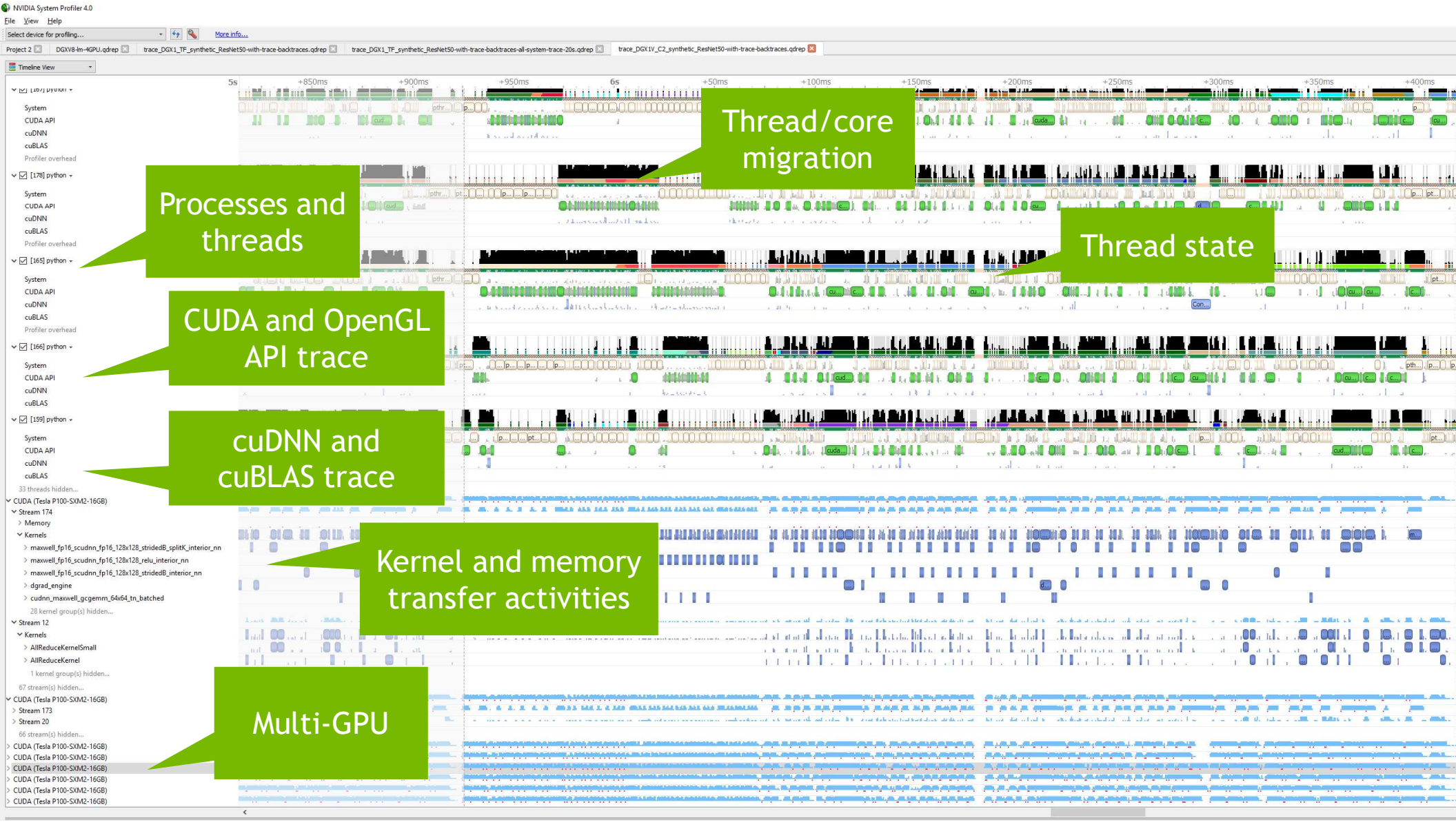
System-wide Performance Analysis

Observe Application Behavior: CPU threads, GPU traces, Memory Bandwidth and more

Locate Optimization Opportunities: CUDA & OpenGL APIs, Unified Memory transfers, User Annotations using NVTX

Ready for Big Data: Fast GUI capable of visualizing in excess of 10 million events on laptops, Container support, Minimum user privileges

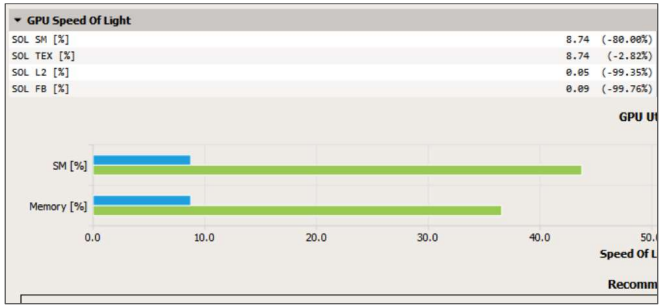
https://developer.nvidia.com/nsight-systems


Thread/core migration
Processes and threads
Thread state
CUDA and OpenGL API trace
cuDNN and cuBLAS trace
Kernel and memory transfer activities
Multi-GPU

NVIDIA NSIGHT COMPUTE

Next Generation Kernel Profiler

- Interactive CUDA API debugging and kernel profiling
- Fast Data Collection
- Improved Workflow and Fully Customizable (Baselining, Programmable UI/Rules)
- Command Line, Standalone, IDE Integration
- Platform Support
  - OS: Linux (x86, ARM), Windows
  - GPUs: Pascal, Volta, Turing



Kernel Profile Comparisons with Baseline

inst_executed [inst]	16,528.00; 16,528.00; …	13,476.00; 13,476.00; …
l1tex__sol_pct [%]	14.33	n/a
launch__block_size	128.00	128.00
launch__function_pcs	47,611,587,968.00	12,273,728.00
launch__grid_size	4,132.00	3,369.00
launch__occupancy_limit_blocks [block]	32.00	32.00
launch__occupancy_limit_registers [register]	21.00	21.00
launch__occupancy_limit_shared_mem [bytes]	384.00	384.00
launch__occupancy_limit_warps [warps]	16.00	16.00
launch__occupancy_per_block_size	3,638.00	3,638.00
launch__occupancy_per_register_count	5,792.00	5,792.00
launch__occupancy_per_shared_mem_size	2,260.00	2,260.00
launch__registers_per_thread [register/thread]	17.00	17.00
launch__shared_mem_config_size [bytes]	49,152.00	49,152.00
launch__shared_mem_per_block_dynamic [bytes/block]	0.00	0.00
launch__shared_mem_per_block_static [bytes/block]	20.00	20.00
launch__thread_count [thread]	528,896.00	431,232.00
launch__waves_per_multiprocessor	3.23	42.11
ltc__sol_pct [%]	6.93	7.18
memory_access_size_type [bytes]	2.00; 32.00; 32.00; 32…	2.00; 32.00; 32.00; 32…

Metric Data

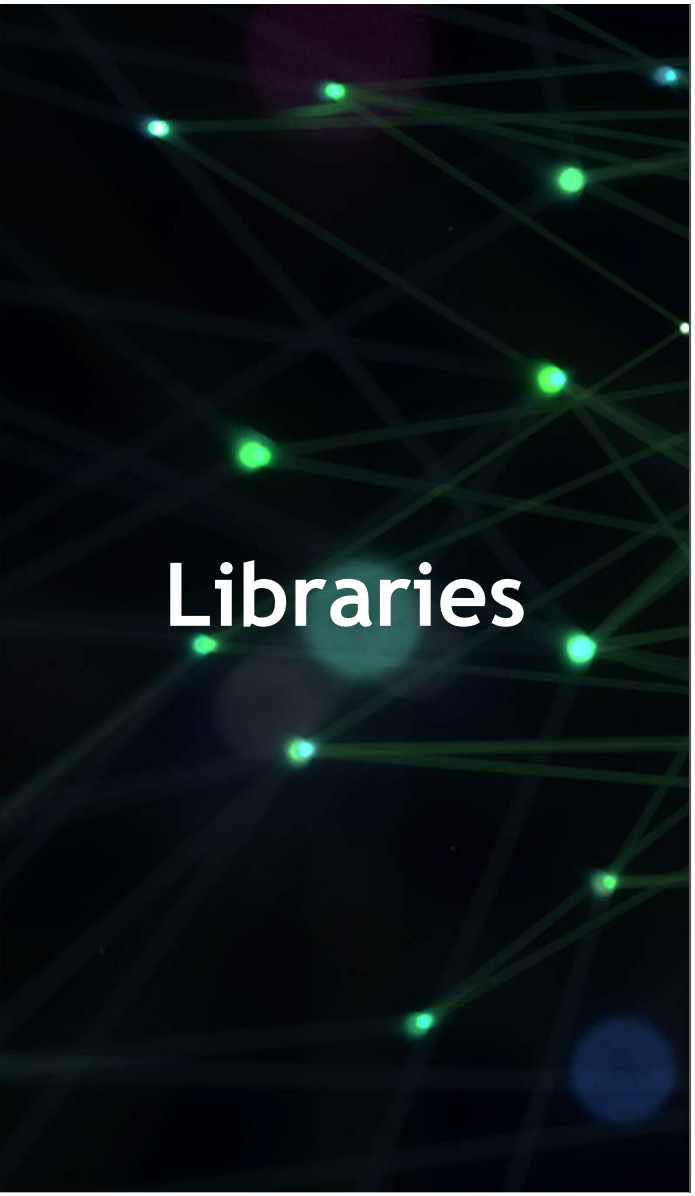
Source	Live Registers	Sampling Data (All)	Sampling Data (No Issue)
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	0	223	0
MOV R1, c[0x0][0x28];	1	13	44
S2R R0, SR_CTAID.X;	2	143	75
S2R R2, SR_TID.X;	3	0	38
IMAD R0, R0, c[0x0][0x0], R2;	3	599	94
ISETP.GE.AND P0, PT, R0, c[0x0][0x170]	2	125	26
@P0 EXIT;	2	259	86
MOV R2, R0;	3	386	29
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	2	0	0
MOV R4, 0x4;	3	0	0
IMAD.WIDE R4, R2, R4, c[0x0][0x160];	4	0	0
LDG.E.SYS R3, [R4];	3	0	0
BSSY B0, 0xb00976780;	3	0	0
SHF.R.S32.HI R0, RZ, 0x1f, R2;	4	0	0

Source Correlation

NSIGHT SYSTEM - DEEP LEARNING FEATURES

Making DL more accessible

- Record DNN graph layers evaluation timing
- Map DNN graph layers evaluation onto CPU & GPU timeline
- NVTX code annotation extensions & GPU mapping
- Programmatic profiler start & stop mechanism
- TensorRT & NCCL trace
- Improved fork and OpenMPI launch support
- Data access mechanisms
- Integration with DLprof/TensorBoard



- Libraries are the quickest path to performance for the largest fraction of codes
- Reduce time to robust performance
- Keep pace with new HW features

Milind

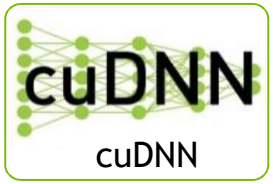
LIBRARIES: EASY, HIGH-QUALITY ACCELERATION

EASE OF USE	• Using libraries enables GPU acceleration without in-depth knowledge of GPU programming
"DROP-IN"	• Many GPU-accelerated libraries follow standard APIs, thus enabling acceleration with minimal code changes
QUALITY	• Libraries offer high-quality implementations of functions encountered in a broad range of applications
PERFORMANCE	• NVIDIA libraries are tuned by experts

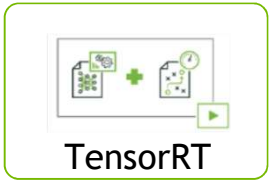
GPU ACCELERATED LIBRARIES

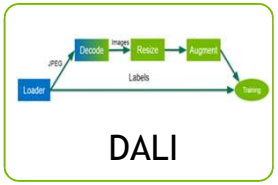
"Drop-in" Acceleration for Your Applications

DEEP LEARNING



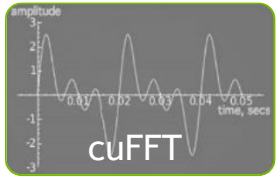
cuDNN


TensorRT



DALI

SIGNAL & IMAGE PROCESSING



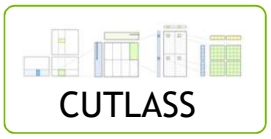
cuFFT


NVIDIA NPP


nvJPEG

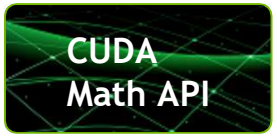
LINEAR ALGEBRA


cuBLAS


CUTLASS

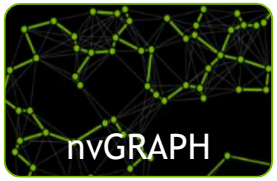

cuSOLVER

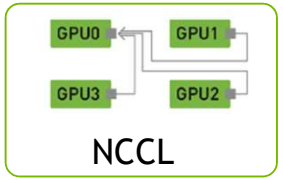

cuSPARSE


CUDA
Math API


cuRAND

PARALLEL ALGORITHMS


nvGRAPH



NCCL

CUDA 9.2 – MATHS HIGHLIGHTS

VOLTA PLATFORM SUPPORT

Volta architecture optimized GEMMs, & GEMM extensions for Volta Tensor Cores (cuBLAS)

Out-of-box performance on Volta (all libraries)

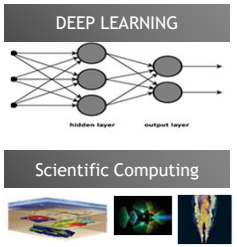
PERFORMANCE

GEMM optimizations for RNNs (cuBLAS)

Faster image processing (NPP)

Prime factor FFT performance (cuFFT)

SpMV performance (cuSPARSE)

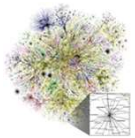


NEW ALGORITHMS

Mixed-precision Batched GEMM for attention models (cuBLAS)

Image Augmentation and batched image processing routines (NPP)

Batched pentadiagonal solver (cuSPARSE)



MEMORY & FOOTPRINT OPTIMIZATION

Large FFT sizes on multi-GPU systems (cuFFT)

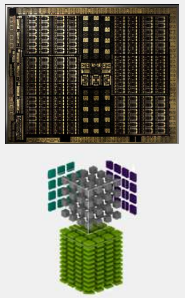
Modular functional blocks with small footprint (NPP)

CUDA 10.0 – MATH LIBRARIES

TURING

Turing optimized GEMMs, & GEMM extensions for Tensor Cores

Turing architecture-optimized libraries

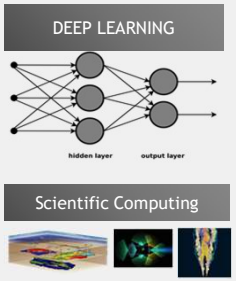


PERFORMANCE

Large FFT & 16-GPU Strong Scaling

Symmetric Eigensolver & Cholesky Performance

cuSPARSE Sparse-Dense Matrix Multiply Performance



NEW ALGORITHMS AND APIs

GPU-accelerated hybrid JPEG decoding

FP16 & INT8 GEMMs for TensorRT Inference

COMPATIBILITY & RELEASE CADENCE

Faster & Independent Library Releases

Library and CUDA compatibility with enterprise drivers

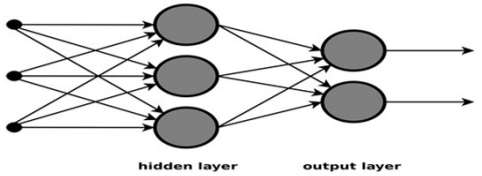
cuBLAS

GPU-accelerated library for dense linear algebra

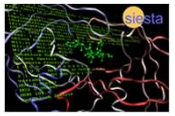
Accelerated library with complete BLAS plus extensions

- Supports all 152 standard routines for single, double, complex, and double complex
- Supports half-precision (FP16), integer (INT8) matrix and mixed precision multiplication operations
- Batched routines for higher performance on small problem sizes
- Host and device-callable interface
- XT interface supports distributed computations across multiple GPUs

DEEP LEARNING
(fully connected layers)



Scientific Computing



SIESTA (MD)

www.flapw.de
Fleur

COSMO, GENE, ELPA…

b-initio
Vienna Ackage Imulation

https://developer.nvidia.com/cublas

cuBLAS 9.2

GEMM Performance for DL RNNs and Convolutional Seq2Seq Models

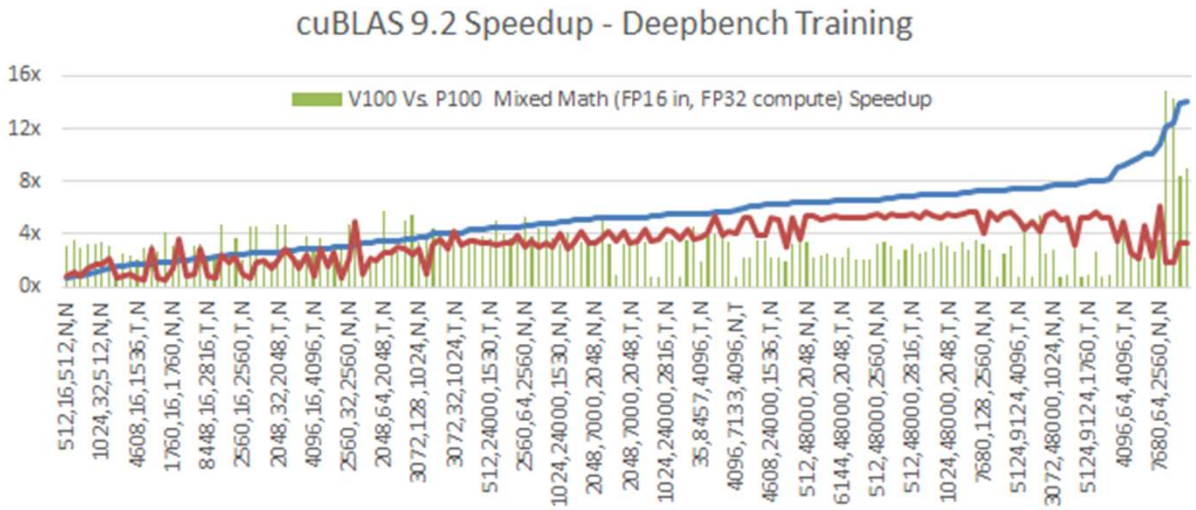
GEMM performance for

- Small tile sizes used in RNN models, Convolutional Seq2seq and OpenAI
- No API changes

Volta architecture optimized heuristics and kernels

API and Error Logging for debug and traceability

5-8x Tensor Op speedup on key DL sizes



https://developer.nvidia.com/cublas

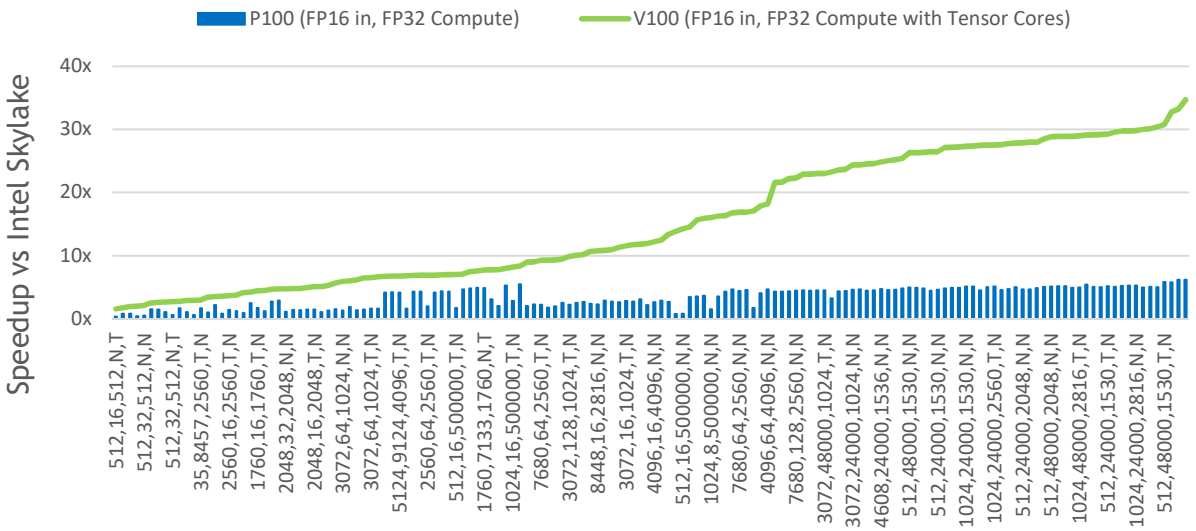
cuBLAS 10.0

Optimized GEMM Performance for Deep Learning

- Turing optimized GEMMs & GEMM extensions for Tensor Cores
- GEMM Performance Tuned for sizes used in various DL models
- API and Error Logging for debug and traceability

https://developer.nvidia.com/cublas

Up to 90TF of Deepbench GEMM Performance



Deepbench training performance runs with Tesla P100, Tesla V100 and Intel Skylake 6140 Gold 2.3 GHz with hyperthreading off

cuBLAS 10.1: New Matrix Multiplication API

Offers future-proofing, flexibility and auto-tuning for GEMM routines

Abstraction of BLAS extensions and special data layouts used in DL applications

- Mixed-precision GEMMs, architecture specific data layouts, complex input/output formats

Supports custom configs & layouts (i.e. workspace) to perform intermediate calculations

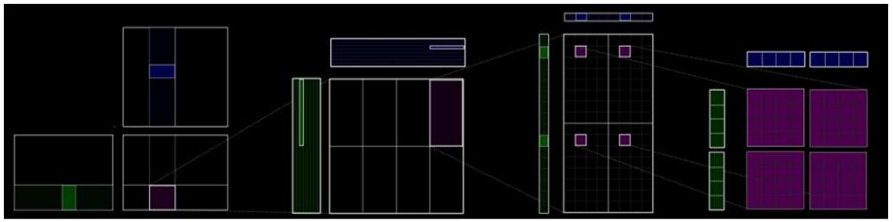
Enables auto-tuning efforts with Find-GEMM API to pick optimal algo for given problem sizes

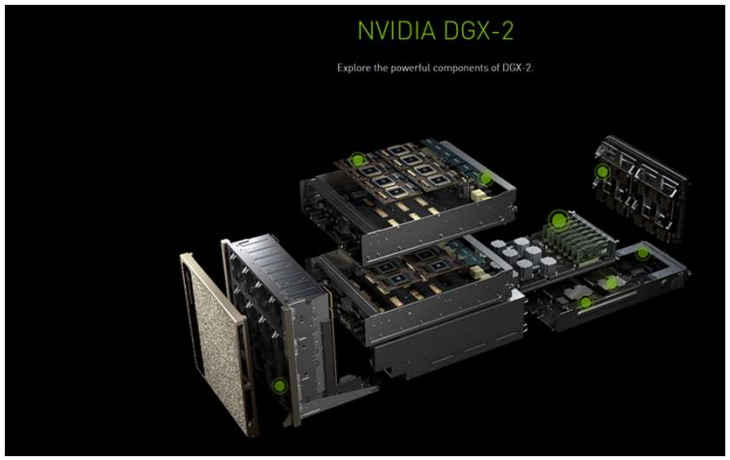
https://developer.nvidia.com/cublas

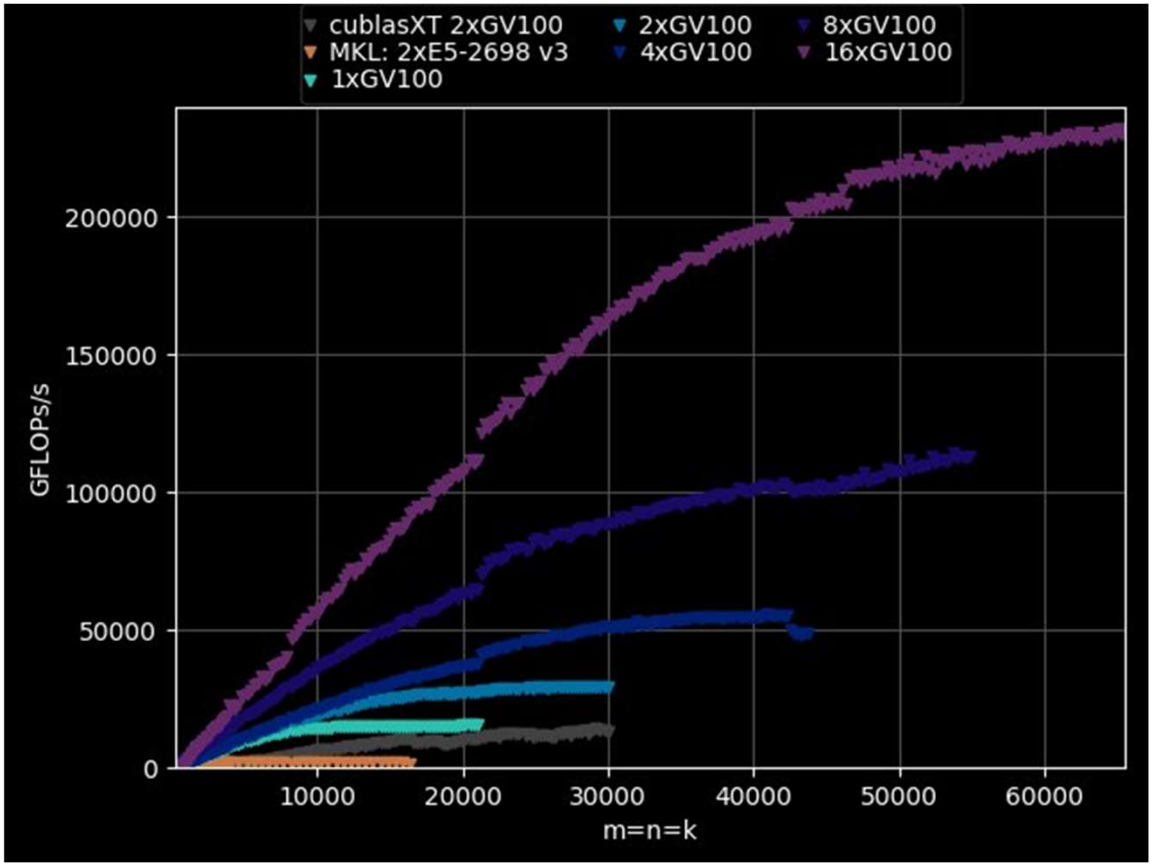
Next: Faster multi-GPU BLAS

cuBLASXT Roadmap: Dense matrix multiplication

$$D = \alpha A \times B + \beta C$$







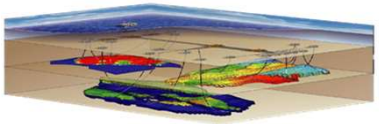
cuFFT

Complete Fast Fourier Transforms Library

Complete Multi-Dimensional FFT Library

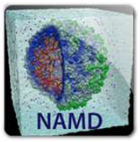
- "Drop-in" replacement for CPU FFTW library
- Real and complex, single- and double-precision data types
- Includes 1D, 2D and 3D batched transforms
- Support for half-precision (FP16) data types
- Supports flexible input and output data layouts
- XT interface now supports up to 8 GPUs

OIL & GAS WELL MODELING



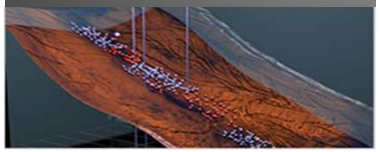
LIFE SCIENCES

GROMACS
FAST. FLEXIBLE. FREE.

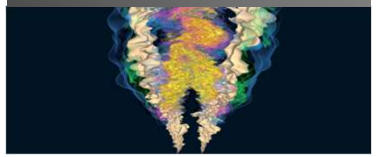


VASP
b-initio
ienna
ackage
imulation

SEISMIC EXPLORATION



COMBUSTION SIMULATION



https://developer.nvidia.com/cufft

cuFFT 9.2

Performance and Memory optimizations

Prime-factor FFT performance with fused Bluestein kernels

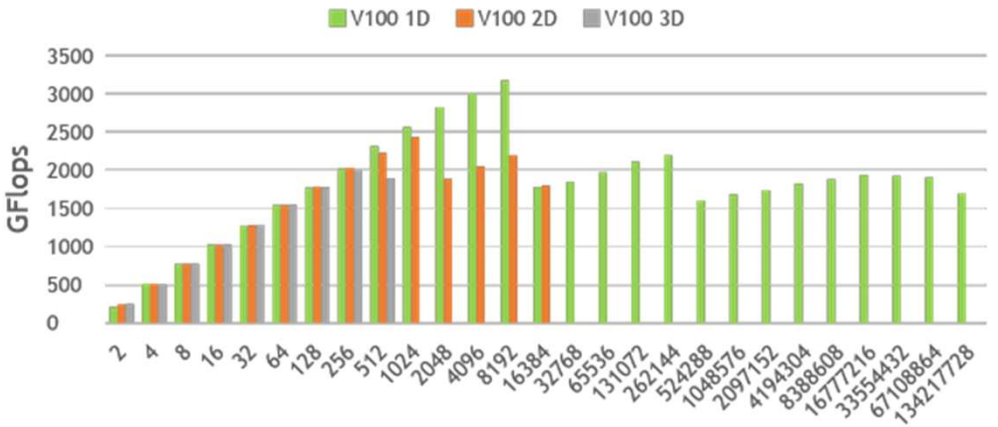
Support for large datasets with new API for low memory usage

Static library without callbacks for datacenter deployments

Maximum cuFFT size

		# of GPUs w/ 16GB RAM			
		1	2	4	8
Dimension	1D	<2^30 (1.0E+9)	<2^30 (1.0E+9)	<2^31 (2.1E+9)	<2^32 (4.2E+9)
	2D	<2^15 (32.6k)	~2^15 (37.7k)	<2^16 (53.3k)	~2^16 (75.4k)
	3D	<2^11 (1.3k)	~2^10 (1.1k)	~2^10 (1.4k)	<2^11 (1.7k)

Faster Image & Signal Processing



* V100 and CUDA 9 (r384); Intel Xeon Broadwell, dual socket, E5-2698 v4@ 2.6GHz, 3.5GHz Turbo with Ubuntu 14.04.5 x86_64 with 128GB System Memory
* P100 and CUDA 8 (r361); For cublas CUDA 8 (r361): Intel Xeon Haswell, single-socket, 16-core E5-2698 v3@ 2.3GHz, 3.6GHz Turbo with CentOS 7.2 x86-64 with 128GB System Memory

https://developer.nvidia.com/cufft

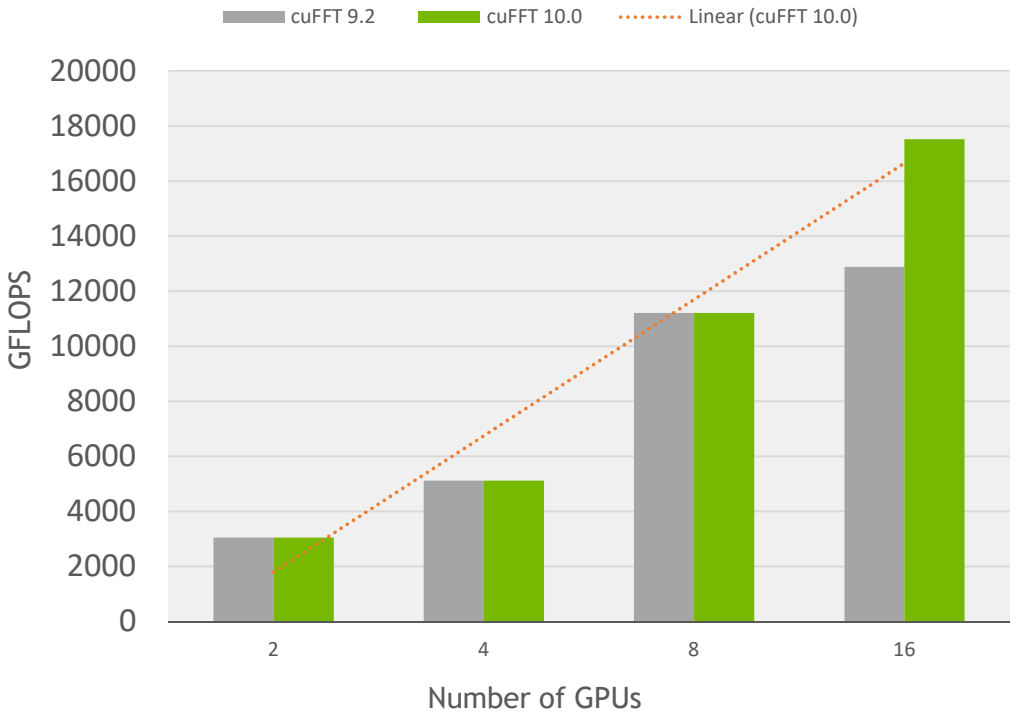
cuFFT 10.0

Multi-GPU Scaling across DGX-2 and HGX-2

- Strong scaling across 16-GPU systems – DGX-2 and HGX-2
- Multi-GPU R2C and C2R support
- Large FFT models across 16-GPUs – effective 512GB vs 32GB capacity

https://developer.nvidia.com/cufft

Up to 17TF performance on 16-GPUs
3D 1K FFT



cuFFT (10.0 and 9.2) using 3D C2C FFT 1024 size on DGX-2

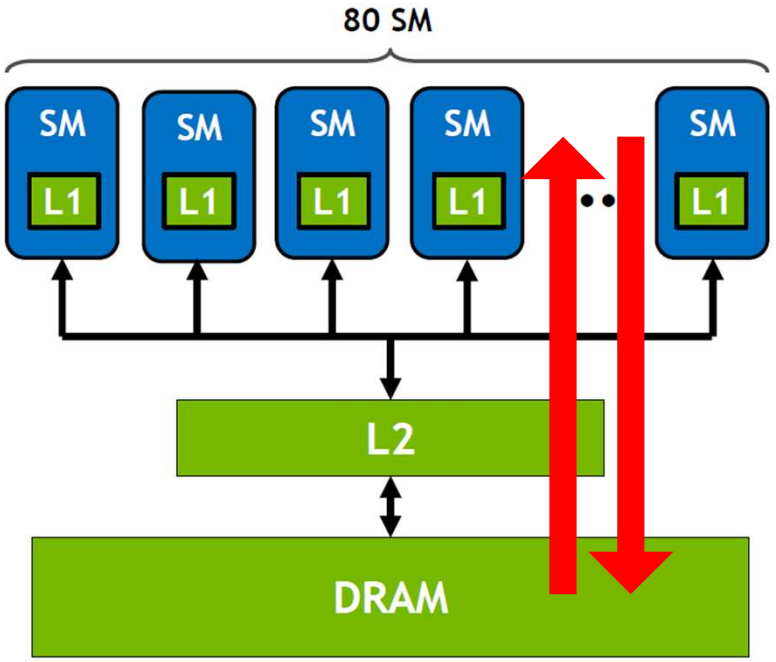
Future: On-chip FFT Library

Device FFT APIs will improve performance, support custom data layouts

Achieves 2x or higher perf by avoiding global memory synchronization

Retains data on to either local or shared memory, allows fusing FFTs with other kernels

Enables customizations on data layouts and operations (convolutions, filtering)

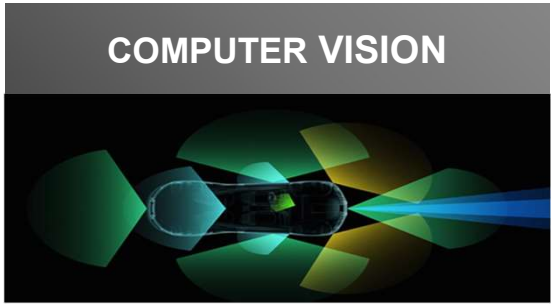


NPP

NVIDIA Performance Primitives Library

GPU-accelerated Building Blocks for Image, Video Processing & Computer Vision

- Over 6000 routines covering image, signal processing and computer vision functions
- 5-100x faster than CPU
- Ease of use: Drop-in to existing project, no GPU kernels required
- Modular building blocks for small memory footprint. Image batching for higher efficiency



https://developer.nvidia.com/npp

cuSPARSE

Sparse Linear Algebra on GPUs

Optimized Sparse Matrix Library

Optimized sparse linear algebra BLAS routines for matrix-vector, matrix-matrix, triangular solve

Support for variety of formats (CSR, COO, block variants)

Incomplete-LU and Cholesky preconditioners

Support for half-precision (fp16) sparse matrix-vector operations

NLP

RECOMMENDATION ENGINES

COMPUTATIONAL FLUID DYNAMICS

SEISMIC EXPLORATION

CAD/CAM/CAE

https://developer.nvidia.com/cusparse

cuSPARSE: Sparse Linear Algebra

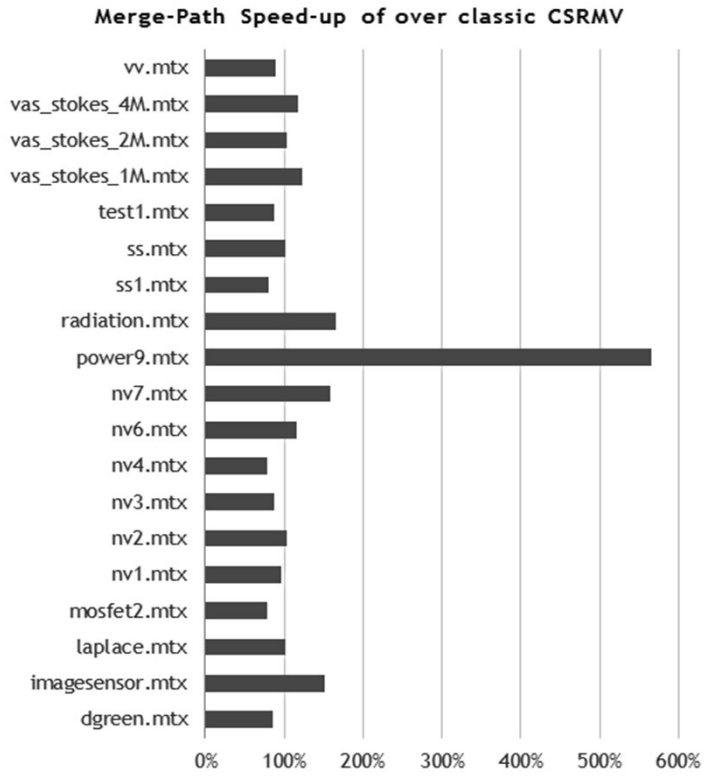
Optimizing Sparse Matrix Multiplication Performance for DL

cuSPARSE 9.2:

- Improved sparse matrix-vector (SpMV) for unstructured matrices with highly variable nnzs per row

Future:

- Focusing on high-performance sparse-dense matrix multiplies (SpMM)
- Also targeting smaller matrices and batched APIs



Benchmarked on GV100 (Volta) GPU using CUDA 9.2

Matrices range from 50k-4.4M size with >99.9 sparsity

Average speed-up of 1.3X

cuSOLVER

Linear Solver Library

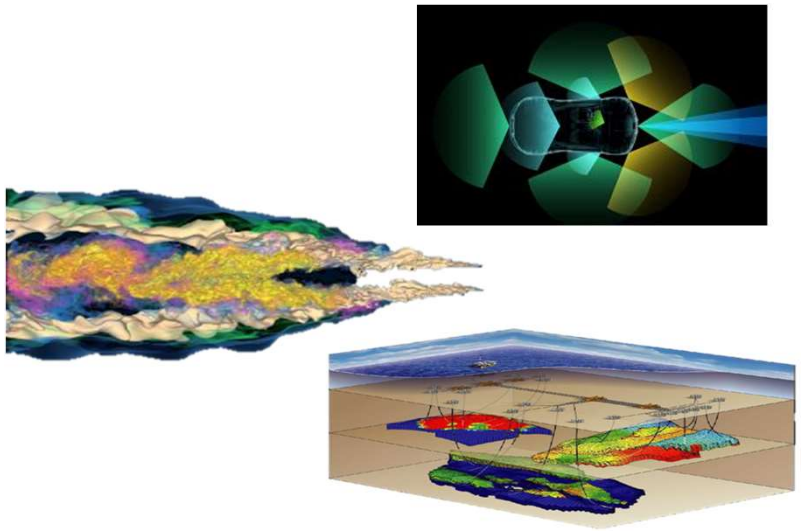
Library for Dense and Sparse Direct Solvers

Supports Dense Cholesky, LU, (batched) QR, SVD and Eigenvalue solvers

Sparse direct solvers & Eigen solvers

Includes a sparse refactorization solver for solving sequences of matrices with a shared sparsity pattern

Used in a variety of applications such as circuit simulation and computational fluid dynamics



Sample Applications

- **Computer Vision**
- **CFD**
- **Newton's method**
- **Chemical Kinetics**
- **Chemistry**
- **ODEs**
- **Circuit Simulation**

https://developer.nvidia.com/cusolver

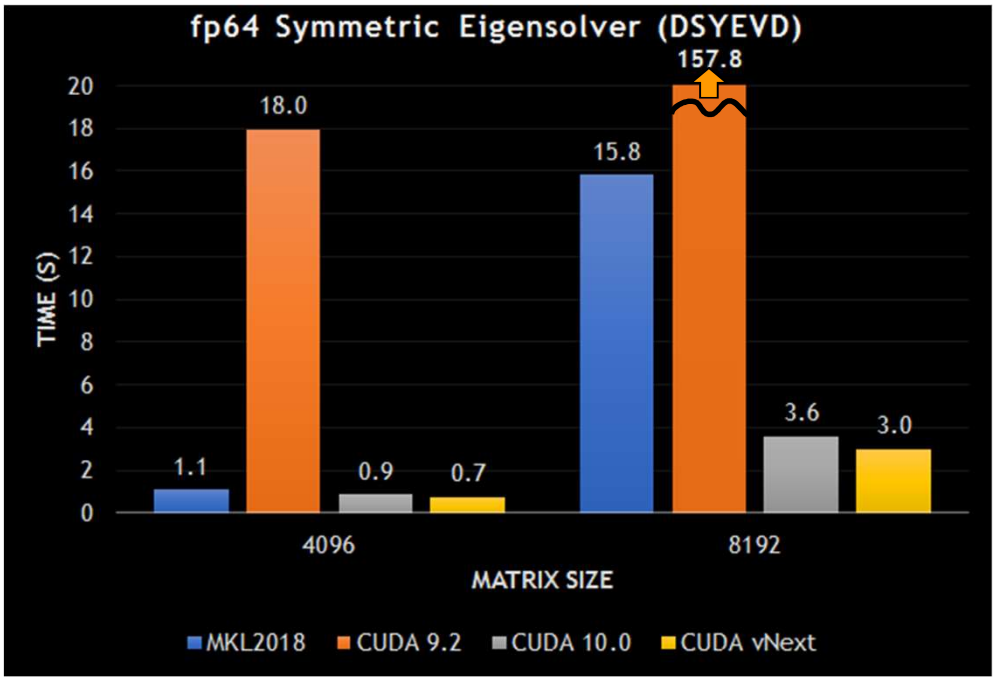
Eigensolver & Cholesky Optimizations

10.0: Improved performance with new implementations for

- Cholesky factorization
- Symmetric & Generalized Symmetric eigensolver
- QR factorization

Future: continue perf tuning & add new functionality:

- Selective eigenvalue/vector solvers
- SVD Perf and batch APIs
- Un-symmetric Eigensolvers



Benchmarks use 2 x Intel Gold 6140 (Skylake) processors and NVIDIA GV100 (Volta) GPUs

Math Libraries 10.0 - Summary

TURING TENSOR CORE 2.0

Turing optimized GEMMs, & GEMM extensions for Tensor Cores 2.0 (cuBLAS, CUTLASS)

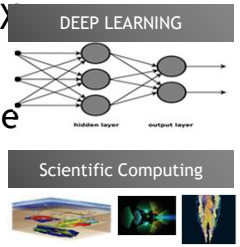
Out-of-box performance on Turing (all libraries)

PERFORMANCE

Large FFT & 16-GPU Perf Scaling on DGX-2/HGX-2 (cuFFT)

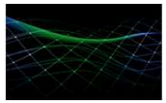
FP16 & INT8 GEMM perf for DL inference (cuBLAS)

Symmetric Eigensolver & Cholesky Perf (cuSOLVER)



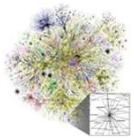
NEW ALGORITHMS & APIs

GPU-accelerated hybrid JPEG decoding (nvJPEG)



New Mat-mul and GEMM Find APIs (cuBLAS)

Mixed-precision batched GEMV, GEMM for Complex data types (cuBLAS)



COMPATIBILITY & RELEASE CADENCE

Faster & Independent Library Releases (starting w/ cuBLAS in Oct, others to follow)

Single library compatible across N and N-1 LTS drivers (r410 and r384)

DEEP LEARNING

Image Classification

Object Detection

COMPUTER VISION

Voice Recognition

Language Translation

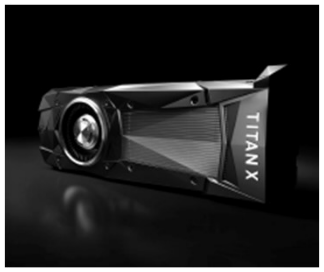
SPEECH AND AUDIO

Recommendation Engines

Sentiment Analysis

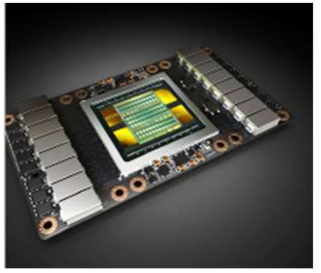
NATURAL LANGUAGE PROCESSING

NVIDIA DEEP LEARNING SOFTWARE TRAINING STACK



At Your Desk

On-Prem



In-the-Cloud

ACCELERATED DEEP LEARNING TRAINING STACK

COMPUTER VISION

SPEECH AND AUDIO

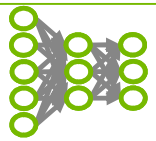
NATURAL LANGUAGE PROCESSING

NGC, GPU Container, KONG, DALI, DIGITS

UI / JOB MANAGEMENT / DATASET VERSIONING/ VISUALIZATION

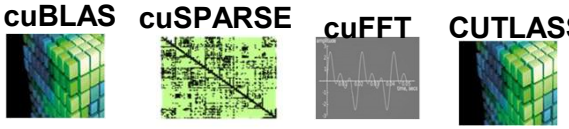
NV OPTIMIZED

NV ACCELERATED



cuDNN

DEEP LEARNING



MATH LIBRARIES

NCCL

COMMUNICATION

NVIDIA cuDNN

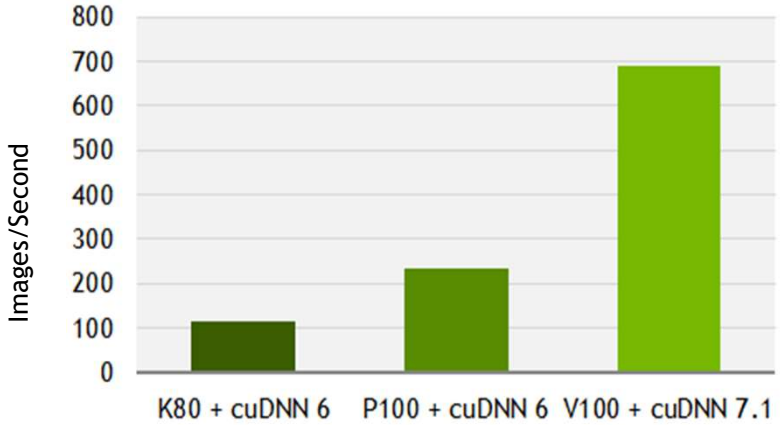
Deep Learning Primitives

High performance building blocks for deep learning frameworks

- Drop-in acceleration for widely used deep learning frameworks
- Accelerates industry vetted deep learning algorithms
- Performance tuned for NVIDIA GPUs

developer.nvidia.com/cudnn

Deep Learning Training Performance



Top: Caffe2 performance (images/sec). Tesla K80 + cuDNN 6 (FP32), Tesla P100 + cuDNN 6 (FP32), Tesla V100 + cuDNN 7.1 (FP16). ResNet50, Batch size: 64

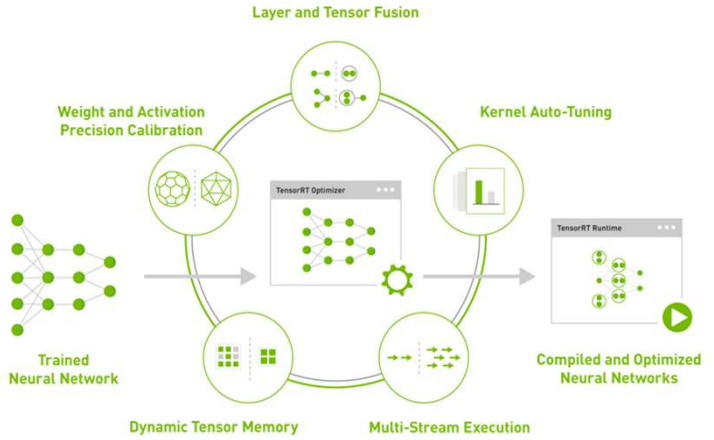
"NVIDIA has improved the speed of cuDNN with each release while extending the interface to more operations and devices at the same time."

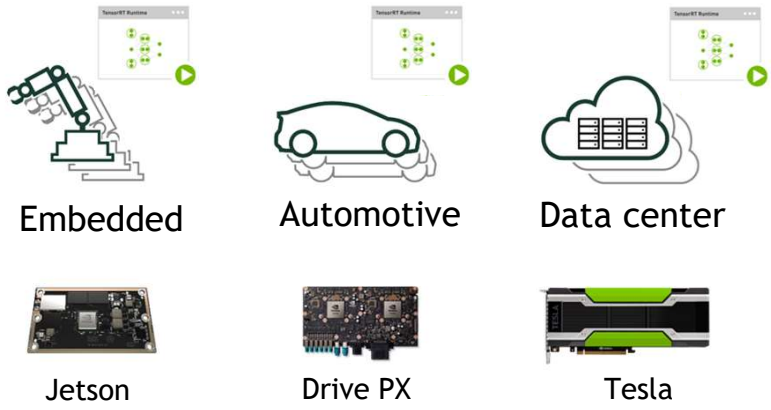
— Evan Shelhamer, Lead Caffe Developer, UC Berkeley

NVIDIA TensorRT

Programmable Inference Accelerator

- Maximize inference throughput for latency-critical services in production environments
- Optimize and deploy models to generate high-performance runtime engines, without the overhead of frameworks
- Deploy faster, more efficient and responsive deep learning applications with INT8 and FP16 precision inference

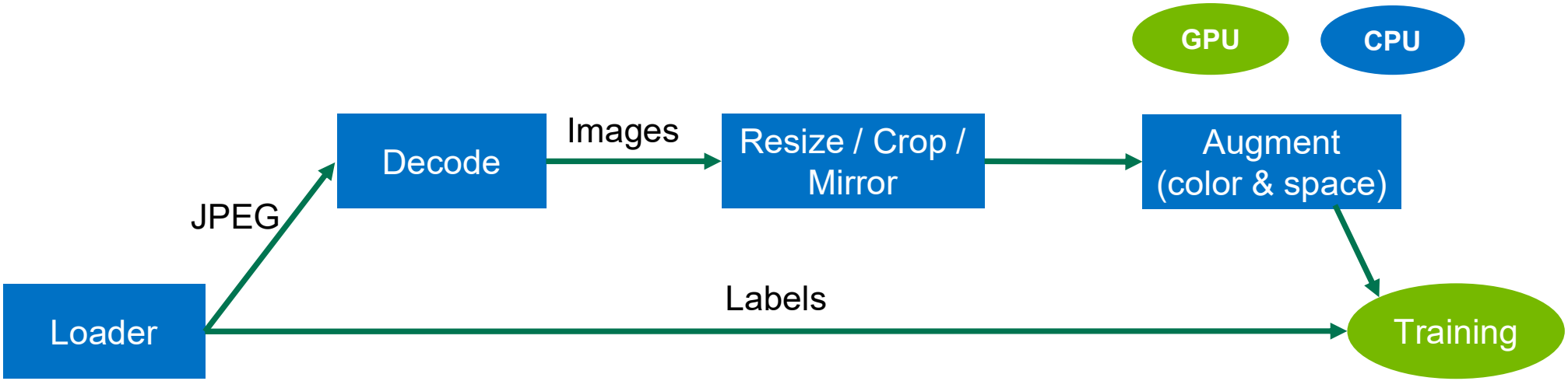




developer.nvidia.com/tensorrt

CPU Bottleneck of DL Training Today

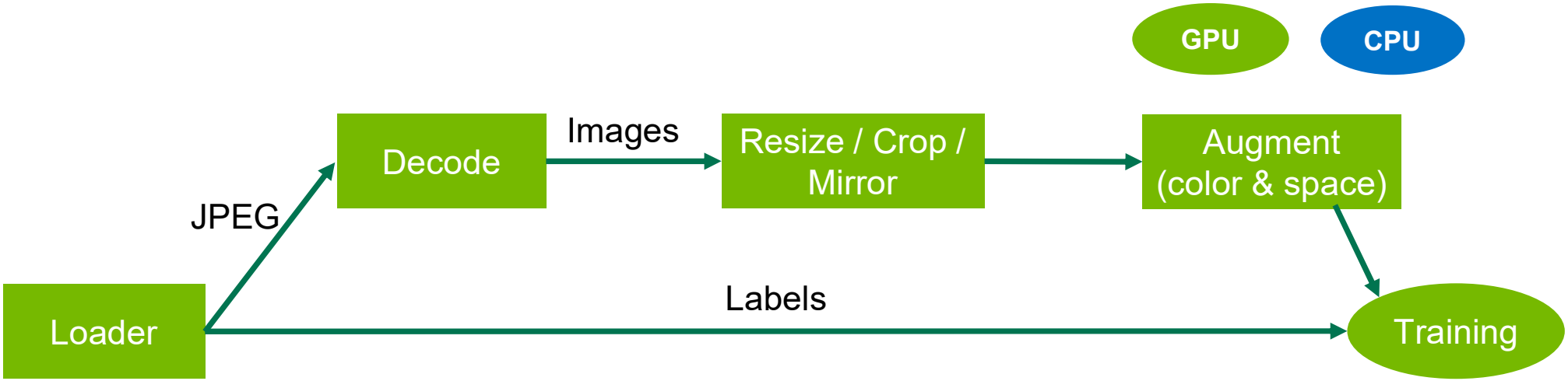
Compute-intensive, multi-stage input pipeline



- **Under-utilized GPU resources. Bottleneck is amplified on multi-GPU systems**
- **Duplicated and sub-optimal pre-processing codes across Frameworks**

Full GPU Pipeline with DALI

Compute-intensive, multi-stage input pipeline



- **Under-utilized GPU resources. Bottleneck is amplified on multi-GPU systems**
- **Duplicated and sub-optimal pre-processing codes across Frameworks**

DALI

Unblock CPU with GPU-accelerated DL pre-processing library

Full input pipeline acceleration including data loading and augmentation

Drop-in integration with direct plugins to frameworks – MxNet, TensorFlow, PyTorch

Portable workflows through multiple input formats and configurable graphs

Available as open source - https://github.com/NVIDIA/DALI

Pre-release v0.1 supports:

- Resnet-50 image classification training
- Input formats – JPEG, LMDB, RecordIO, TFRecord
- Python APIs to define, build and run an input pipeline

(VERY) USEFUL RESOURCE

https://devblogs.nvidia.com/