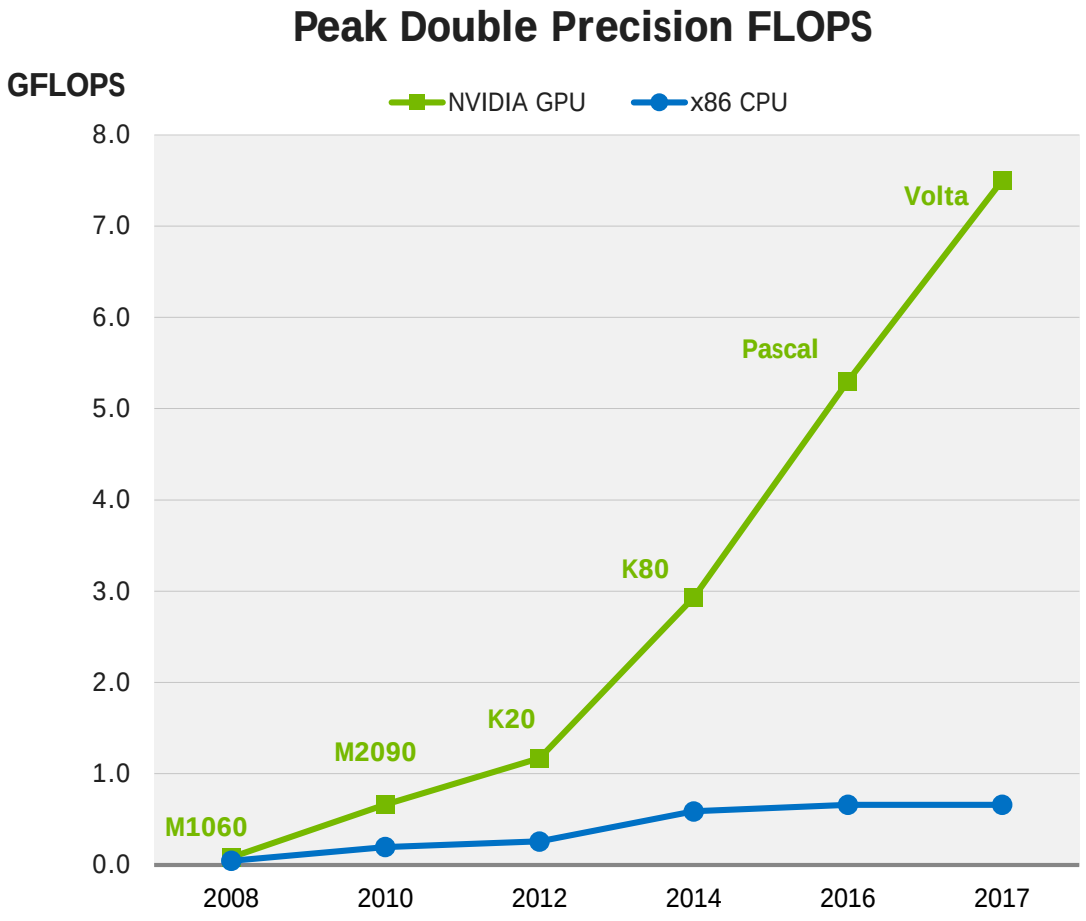
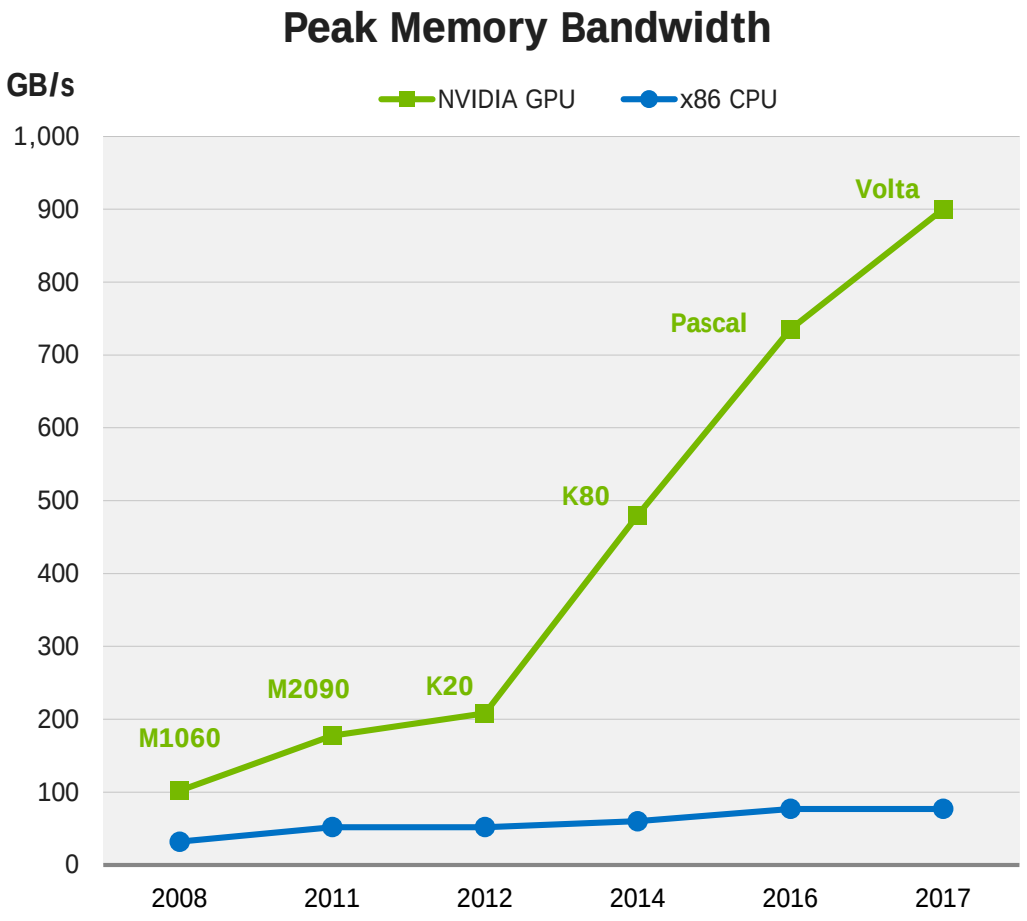
HPC | VOLTA

Ken Hester, Solution Architect

NVIDIA

THE ADVANTAGES OF GPU-ACCELERATED DATA CENTER

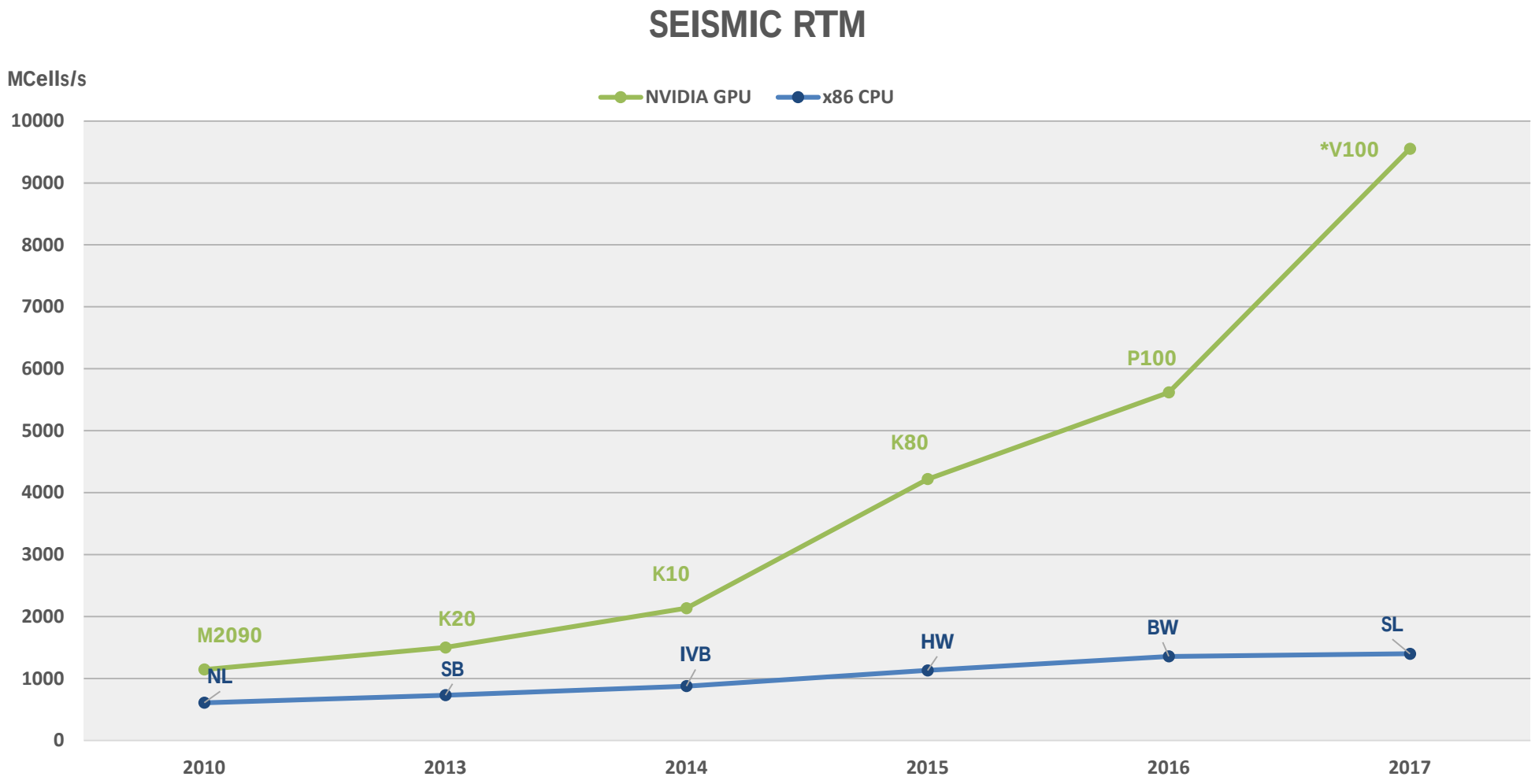




GPU PERFORMANCE COMPARISON

	P100	V100	Ratio
Training acceleration	10 TOPS	120 TOPS	**12x**
Inference acceleration	21 TFLOPS	120 TOPS	**6x**
FP64/FP32	5/10 TFLOPS	7.5/15 TFLOPS	**1.5x**
HBM2 Bandwidth	720 GB/s	900 GB/s	**1.2x**
NVLink Bandwidth	160 GB/s	300 GB/s	**1.9x**
L2 Cache	4 MB	6 MB	**1.5x**
L1 Caches	1.3 MB	10 MB	**7.7x**

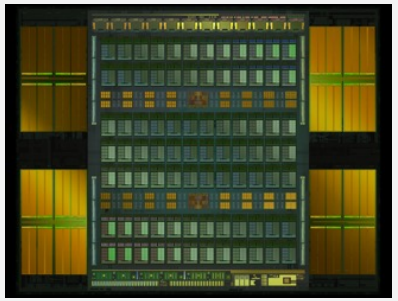
THE ADVANTAGES OF GPU-ACCELERATED DATA CENTER



2 Pass 8th Spatial Order TTI

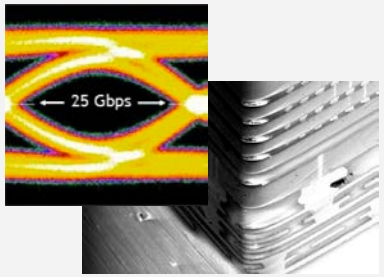
INTRODUCING TESLA V100

Volta Architecture



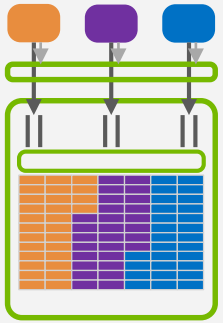
Most Productive GPU

Improved NVLink & HBM2



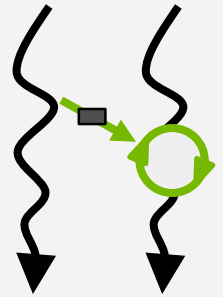
Efficient Bandwidth

Volta MPS



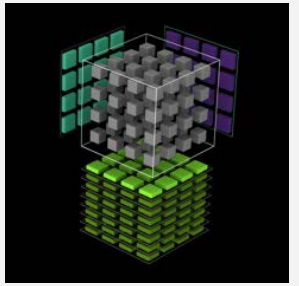
Inference Utilization

Improved SIMT Model



New Algorithms

Tensor Core



120 Programmable TFLOPS Deep Learning

The Fastest and Most Productive GPU for Deep Learning and HPC

TESLA V100

THE MOST ADVANCED DATA CENTER GPU EVER BUILT

5,120 CUDA cores
640 NEW Tensor cores
7.5 FP64 TFLOPS | 15 FP32 TFLOPS
120 Tensor TFLOPS
20MB SM RF | 16MB Cache | 16GB HBM2 @ 900 GB/s
300 GB/s NVLink

TESLA V100

THE MOST ADVANCED DATA CENTER GPU EVER BUILT

5,120 CUDA cores
640 NEW Tensor cores
7 FP64 TFLOPS | 14 FP32 TFLOPS
120 Tensor TFLOPS
20MB SM RF | 16MB Cache | 16GB HBM2 @ 900 GB/s
250W

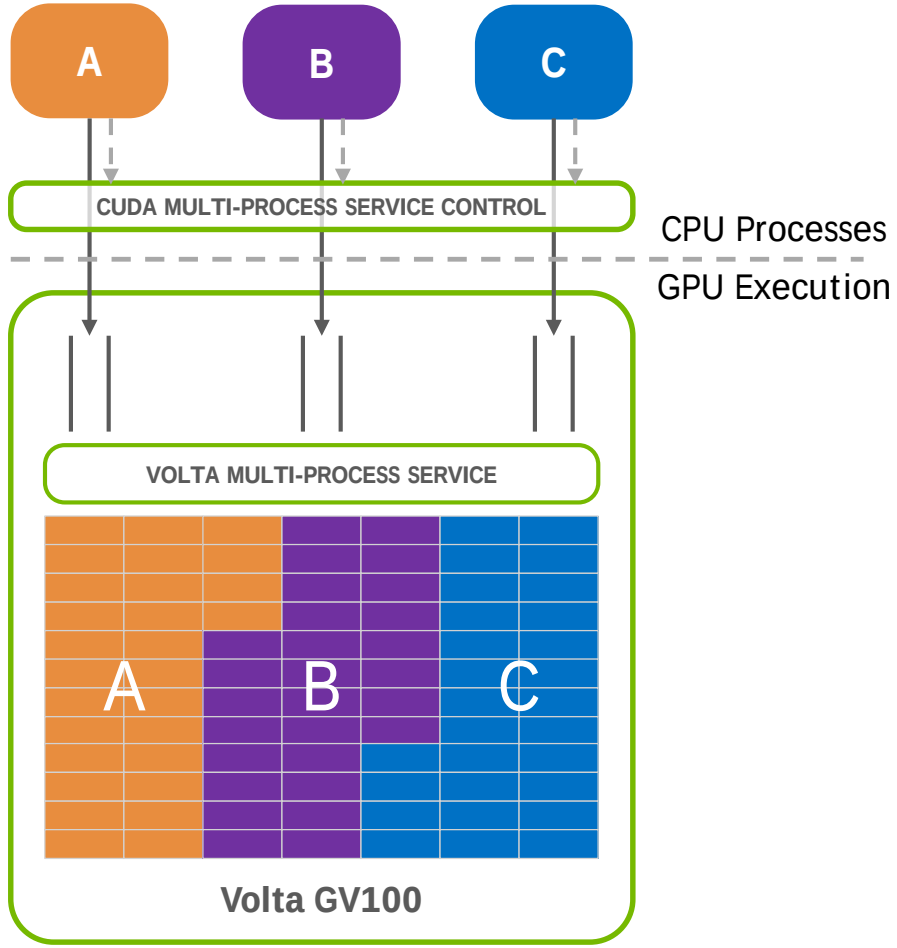
VOLTA MULTI-PROCESS SERVICE

Volta MPS Enhancements:

- Reduced launch latency
- Improved launch throughput
- Improved quality of service with scheduler partitioning
  - More reliable performance
- 3x more clients than Pascal

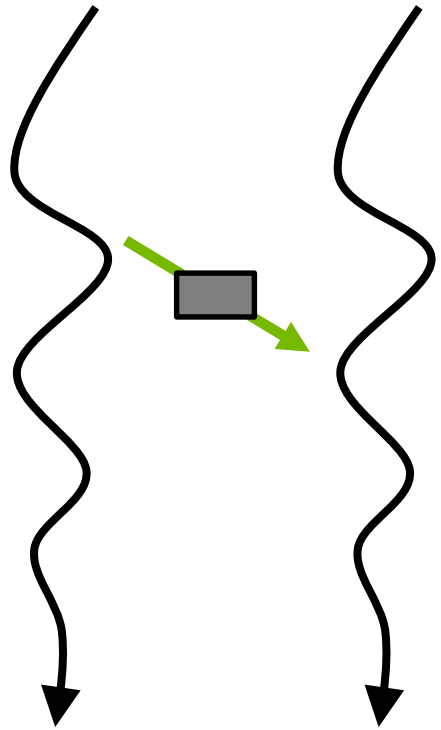
Hardware Accelerated Work Submission

Hardware Isolation



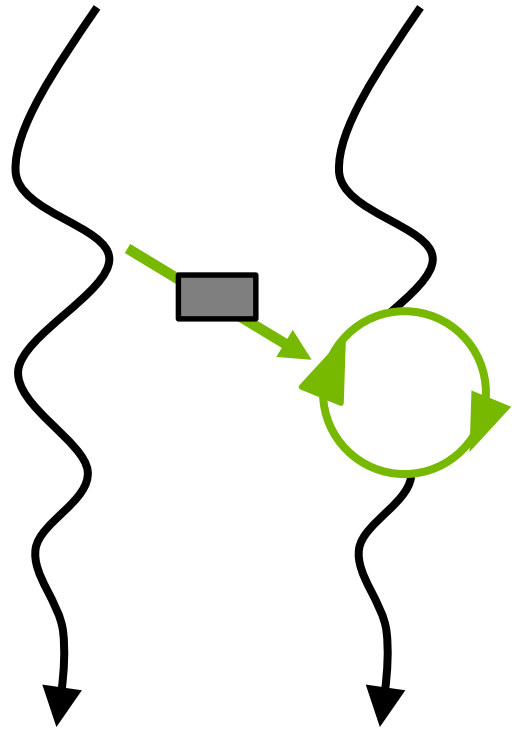
VOLTA: INDEPENDENT THREAD SCHEDULING

Communicating Algorithms



Pascal: Lock-Free Algorithms

Threads cannot wait for messages



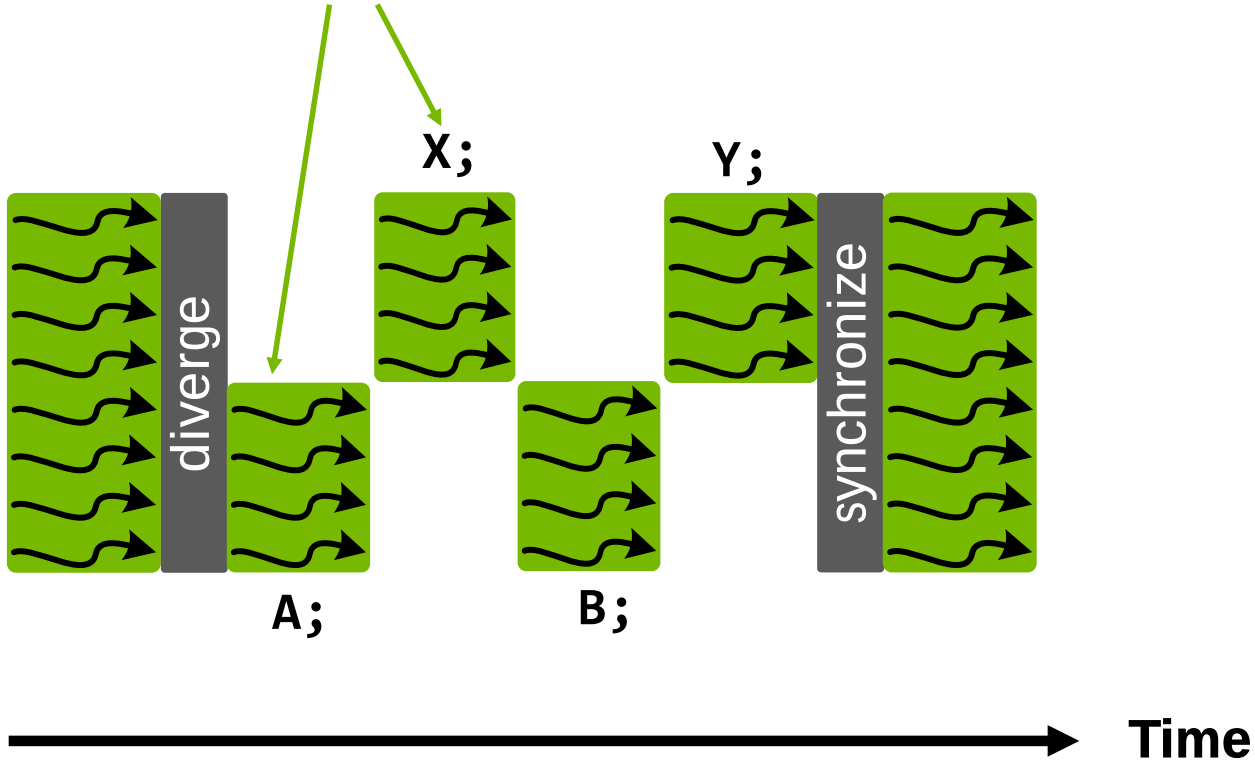
Volta: Starvation Free Algorithms

Threads may wait for messages

VOLTA WARP EXECUTION MODEL

Synchronization may lead to interleaved scheduling!

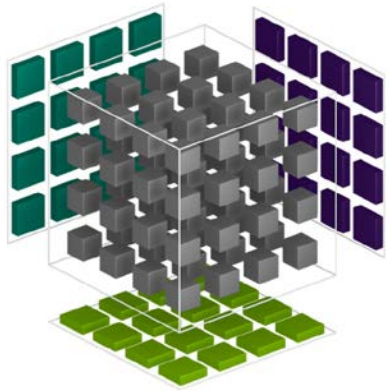
```
if (threadIdx.x < 4) {
    A;
    __syncwarp();
    B;
} else {
    X;
    __syncwarp();
    Y;
}
__syncwarp();
```



Software synchronization also supported, e.g. locks for doubly-linked list!

TENSOR CORE

Mixed Precision Matrix Math
4x4 matrices

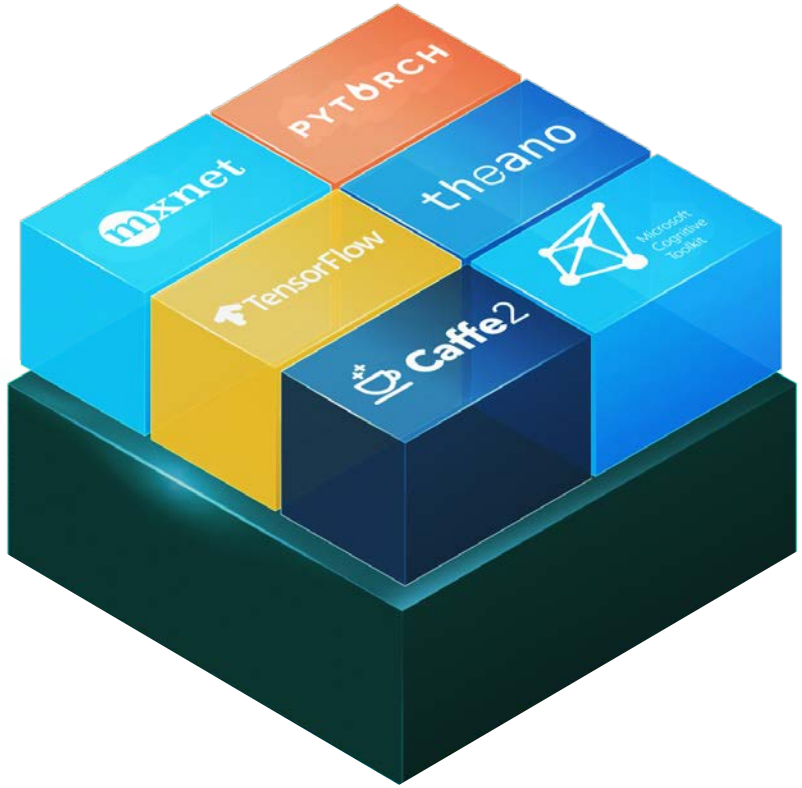


$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

FP16 or FP32 | FP16 | FP16 | FP16 or FP32

$$D = AB + C$$

USING TENSOR CORES



NVIDIA cuDNN, cuBLAS, TensorRT

Volta Optimized Frameworks and Libraries

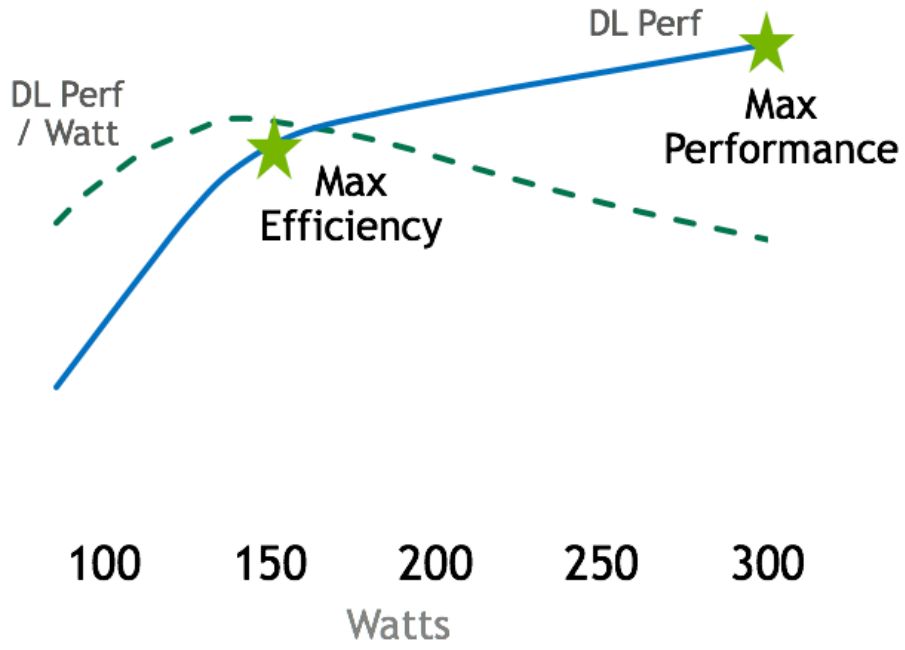
```
__device__ void tensor_op_16_16_16(
    float *d, half *a, half *b, float *c)
{
  wmma::fragment<matrix_a, …> Amat;
  wmma::fragment<matrix_b, …> Bmat;
  wmma::fragment<matrix_c, …> Cmat;

  wmma::load_matrix_sync(Amat, a, 16);
  wmma::load_matrix_sync(Bmat, b, 16);
  wmma::fill_fragment(Cmat, 0.0f);

  wmma::mma_sync(Cmat, Amat, Bmat, Cmat);

  wmma::store_matrix_sync(d, Cmat, 16,
                          wmma::row_major);
}
```

CUDA C++

Warp-Level Matrix Operations

MAXQ: OPTIMIZED FOR DATACENTER EFFICIENCY

40% More Performance in a Rack



80% Perf at Half the Power

V100 Max Performance	V100 Max Efficiency
13 KW Rack 4 Nodes of 8xV100	13 KW Rack 7 Nodes of 8xV100
13	18
ResNet-50 Networks Trained Per Day	ResNet-50 Networks Trained Per Day

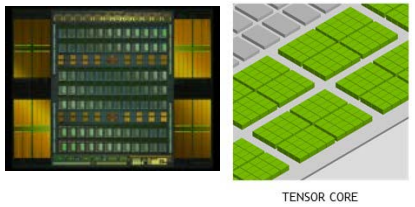
ResNet-50 Training, Max Efficiency run with V100@160W | V100 performance measured on pre-production hardware.

CUDA TOOLKIT 9

Download Now: https://developer.nvidia.com/cuda-toolkit/whatsnew

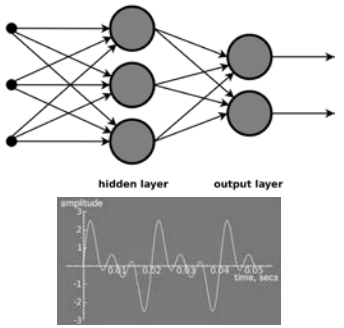
BUILT FOR VOLTA

New GPU Architecture
Tensor Cores
Second-Generation NVLink
HBM2 Stacked Memory



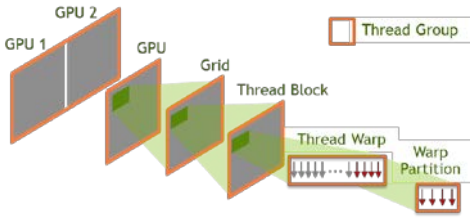
FASTER LIBRARIES

GEMM Optimizations for RNNs (cuBLAS)
>20x Faster Image Processing (NPP)
FFT Optimizations Across Various Sizes (cuFFT)



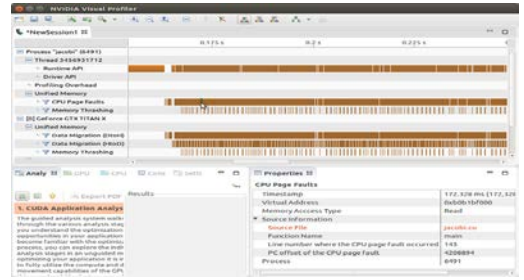
COOPERATIVE THREAD GROUPS

Flexible Thread Groups
Efficient Parallel Algorithms
Synchronize Across Thread Blocks in a Single GPU or Multi-GPUs



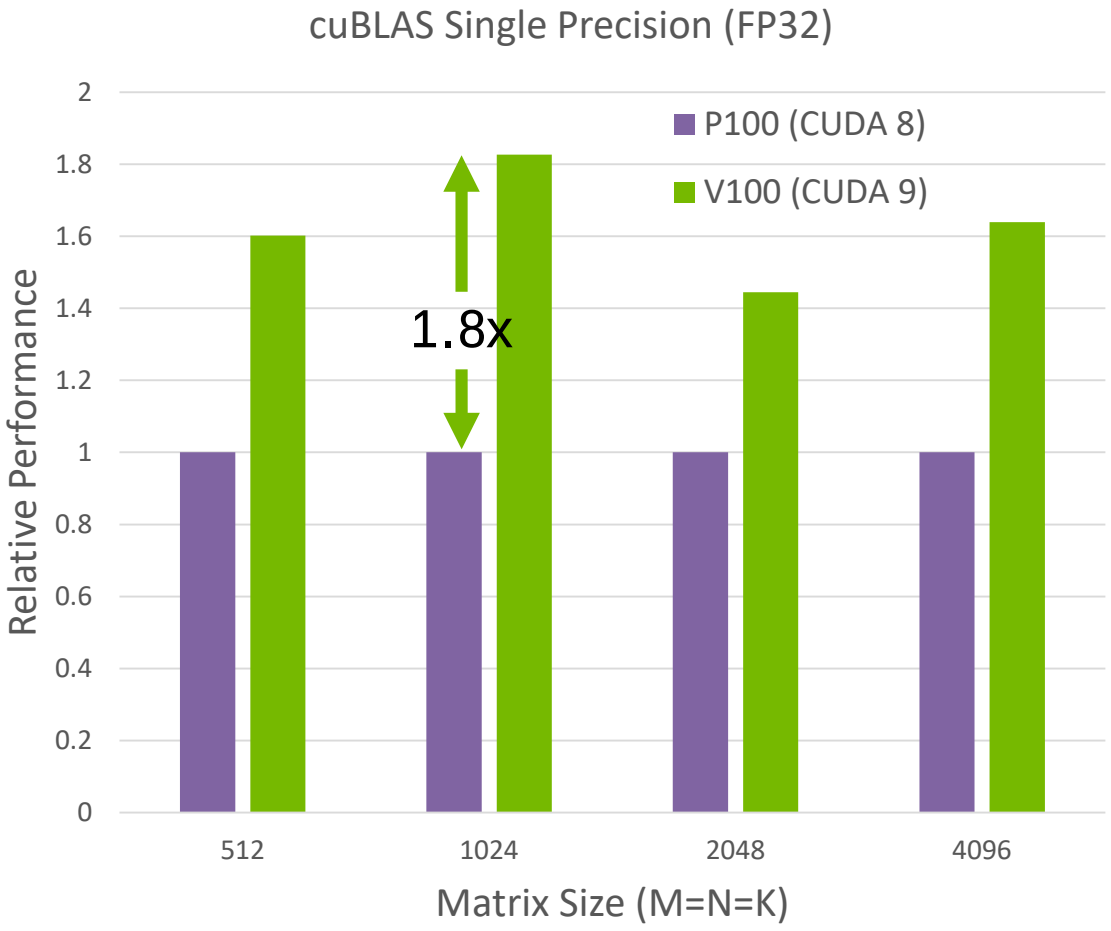
DEVELOPER TOOLS & PLATFORM UPDATES

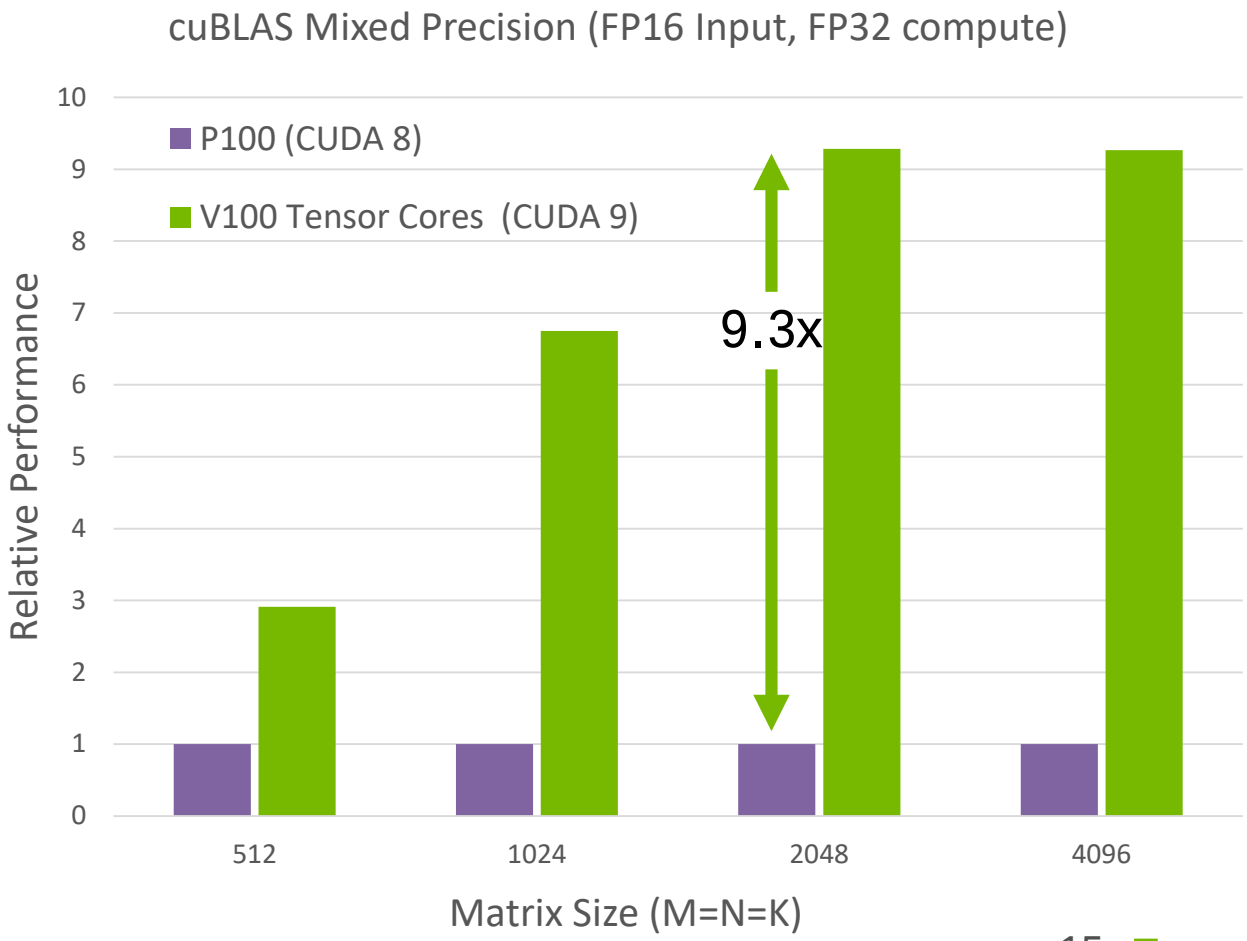
1.3x Faster Compile Times
Unified Memory Profiling
NVLink Visualization
New OS and Compiler Support



CUDA TOOLKIT 9

Download Now: https://developer.nvidia.com/cuda-toolkit/whatsnew





Note: pre-production Tesla V100 and pre-release CUDA 9. CUDA 8 GA release.

PGI 17.7 NEW FEATURES

Download Now: http://www.pgicompilers.com/products/new-in-pgi.htm

- Tesla V100 GPU support
- OpenACC for CUDA Unified Memory
- OpenMP 4.5 for multicore CPUs
- C++14 lambdas with capture in OpenACC regions
- OpenACC PGI Unified Binary for Multicore and Tesla
- C++ performance optimizations
- OpenACC Fortran deep copy

OPENACC IS FOR MULTICORE CPUS & GPUS

```
 98 !$ACC KERNELS
 99 !$ACC LOOP INDEPENDENT
100       DO k=y_min-depth,y_max+depth
101 !$ACC LOOP INDEPENDENT
102         DO j=1,depth
103           density0(x_min-j,k)=left_density0(left_xmax+1-j,k)
104         ENDDO
105       ENDDO
106 !$ACC END KERNELS
```

CPU

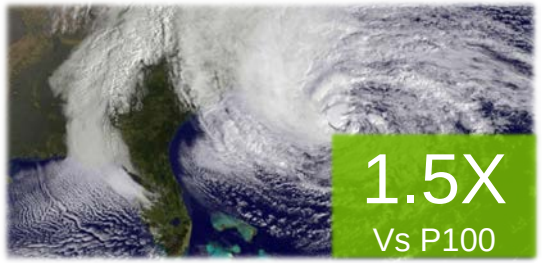
```
% pgfortran -ta=multicore –fast –Minfo=acc -c \
  update_tile_halo_kernel.f90
  . . .
  100, Loop is parallelizable
       Generating Multicore code
       100, !$acc loop gang
  102, Loop is parallelizable
```

GPU

```
% pgfortran -ta=tesla –fast -Minfo=acc –c \
  update_tile_halo_kernel.f90
  . . .
  100, Loop is parallelizable
  102, Loop is parallelizable
       Accelerator kernel generated
       Generating Tesla code
       100, !$acc loop gang, vector(4) ! blockidx%y threadidx%y
       102, !$acc loop gang, vector(32) ! blockidx%x threadidx%x
```

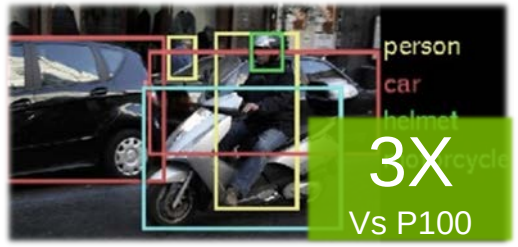
Single Universal GPU For all accelerated workloads

BOOSTS ALL ACCELERATED WORKLOADS



HPC

AI Training



AI Inference

Virtual Desktop

V100 UNIVERSAL GPU

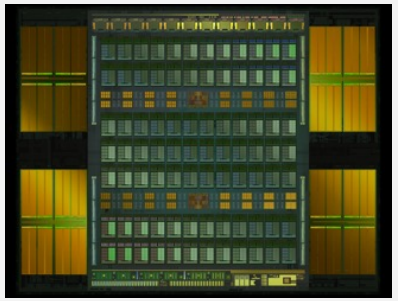
Tesla Products Segmentation

	HYPERSCALE HPC	MULTI-APP HPC	STRONG-SCALE HPC
Products	P40, P4	P100, V100	P100, V100 (NVLINK)
Value Prop	Best perf/$, optimized for hyperscale deployment	Dramatically lowers cost for HPC datacenter	Highest absolute performance
Target Segments	• Deep Learning Training (P40) • Deep Learning Inference (P40/P4) • Video & Image Processing	• HPC Datacenter (mixed CPU/GPU) • Oil & Gas • Climate & Weather • Data Analytics (Database, BI, Visualize)	• Deep Learning Training • Super-computing (Phy, MD, QC, CFD)* • Defense (Graph Analytics, FFT) • Data Analytics (Database, BI, Visualize)
Recommended Configs.	4-8 GPU/node (Training) 1-8 GPU/node (Inference)	2-8 GPU/node	4-8 GPU/node

*Super-computing includes Physics, Molecular Dynamics (MD), Quantum Chemistry (QC), Computational Fluid Dynamics (CFD)

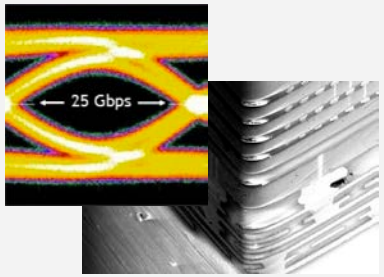
Introducing Tesla V100

Volta Architecture



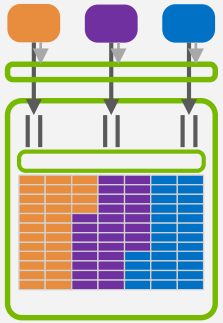
Most Productive GPU

Improved NVLink & HBM2



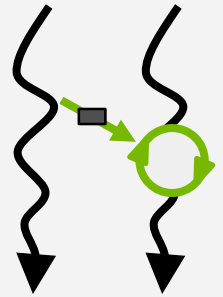
Efficient Bandwidth

Volta MPS



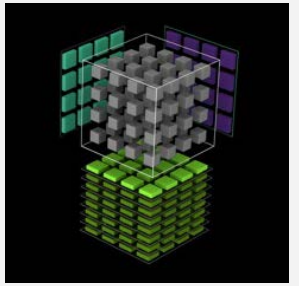
Inference Utilization

Improved SIMT Model



New Algorithms

Tensor Core



120 Programmable TFLOPS Deep Learning

More V100 Features: 2x L2 atomics, int8, new memory model, copy engine page migration, and more …

The Fastest and Most Productive GPU for Deep Learning and HPC

Thank You