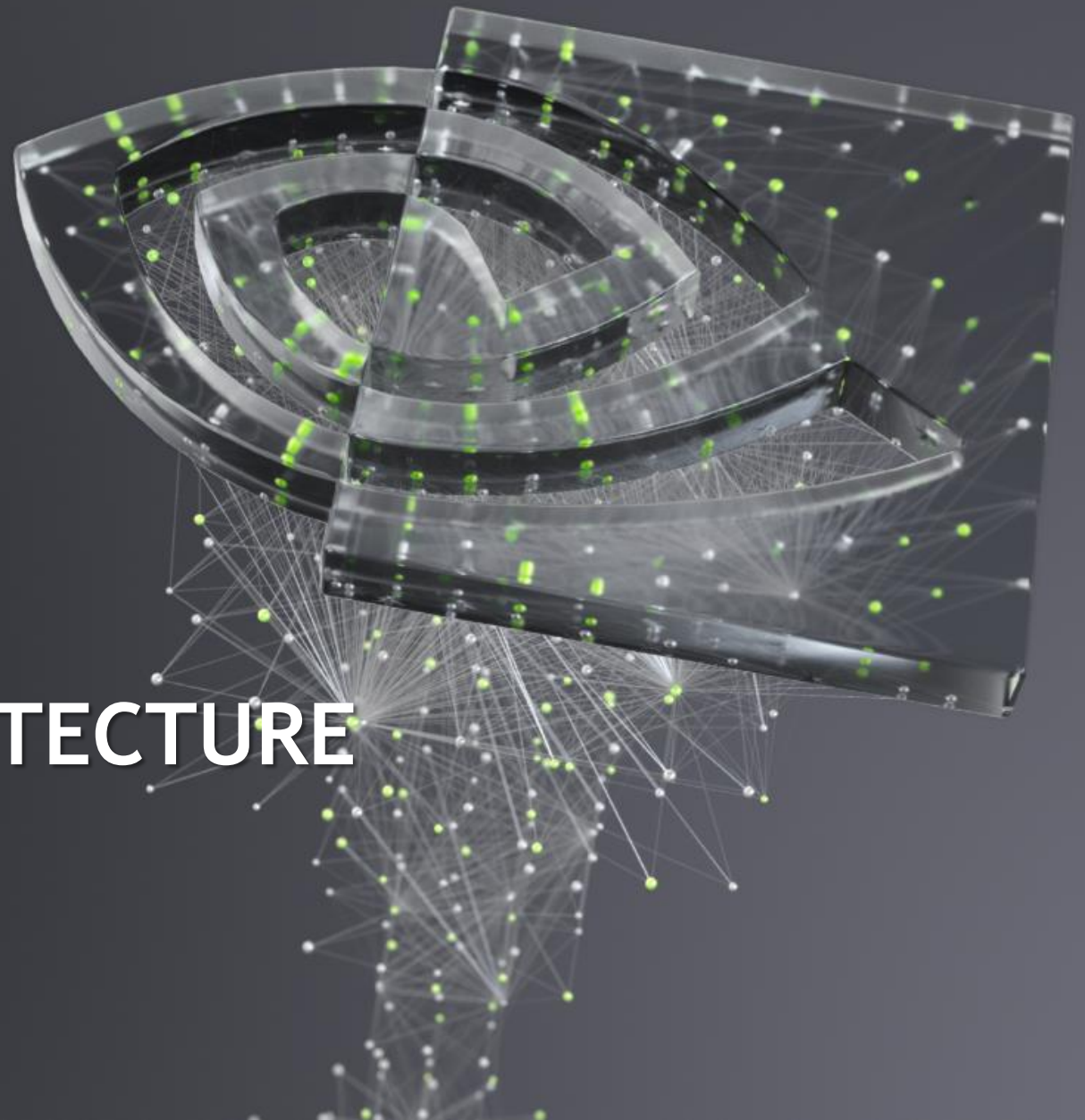


NVIDIA CUDA ARCHITECTURE

João Paulo Navarro – Solutions Architect
jpnavarro@nvidia.com

- NVIDIA history
- Why GPU and accelerated computing
- GPU architecture
- GPU data-center architecture

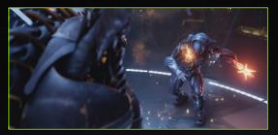
NVIDIA

GRAPHICS

GAMING

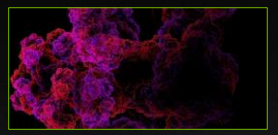
DESIGN

RENDERING



HPC

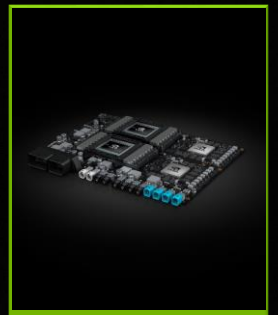
SUPERCOMPUTING



AI

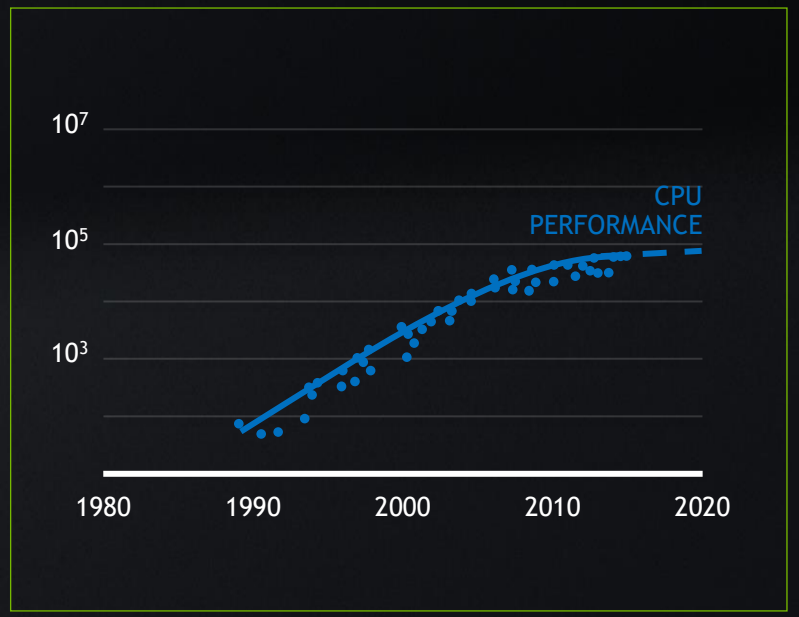
AI TRAINING

AI INFERENCE



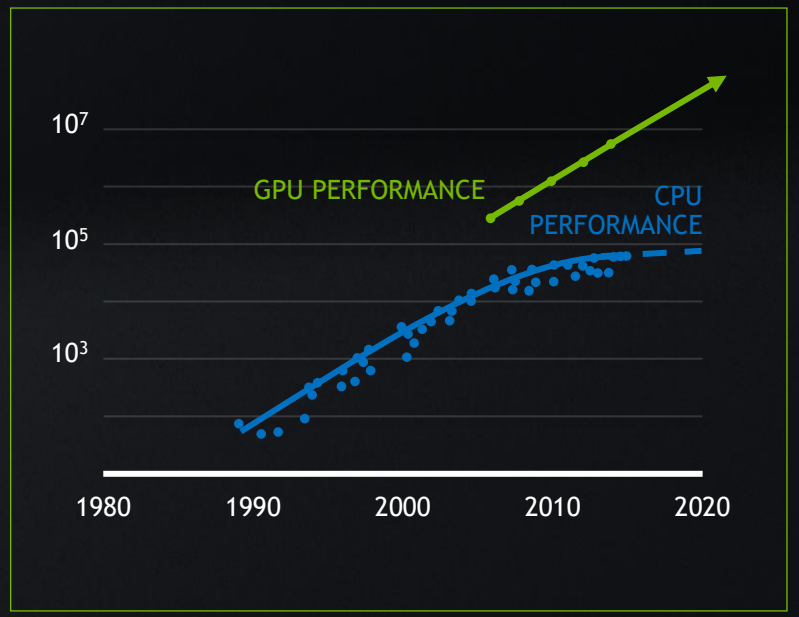
ROBOTICS

FORCES SHAPING COMPUTING

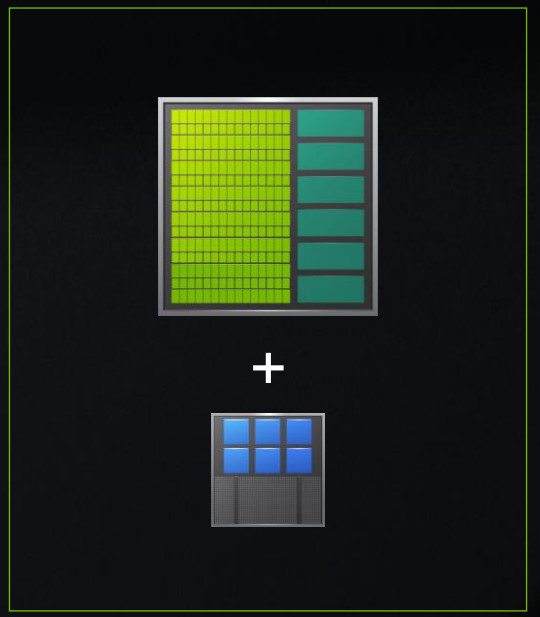


BEYOND MOORE'S LAW

FORCES SHAPING COMPUTING

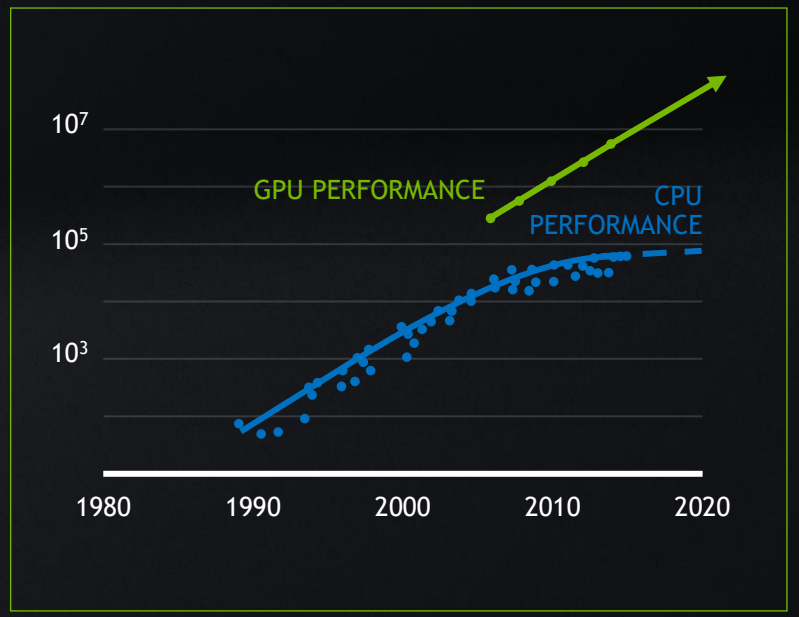


BEYOND MOORE'S LAW – 1000X EVERY 10 YEARS

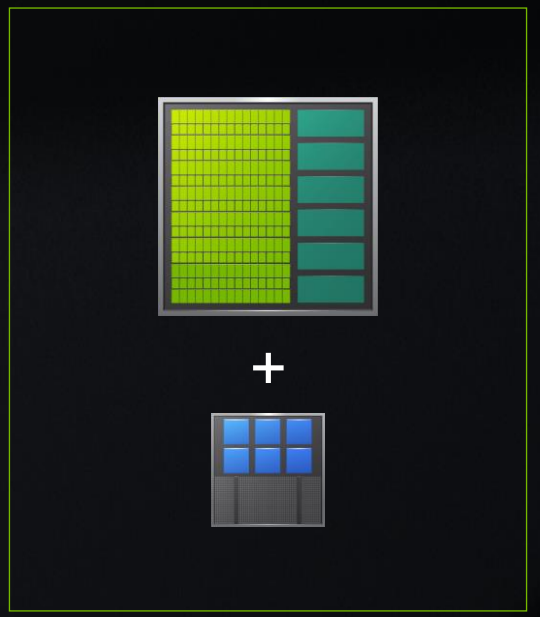


ACCELERATED COMPUTING

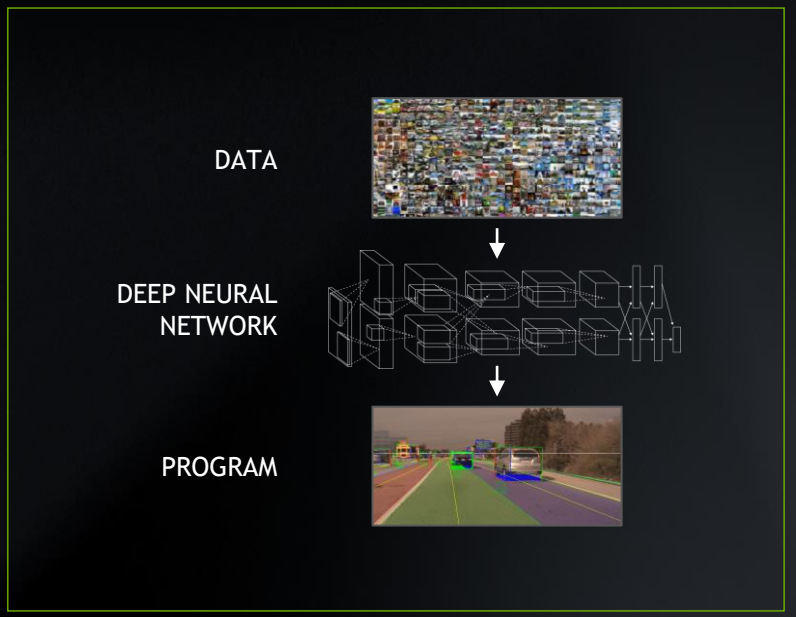
FORCES SHAPING COMPUTING



BEYOND MOORE'S LAW – 1000X EVERY 10 YEARS

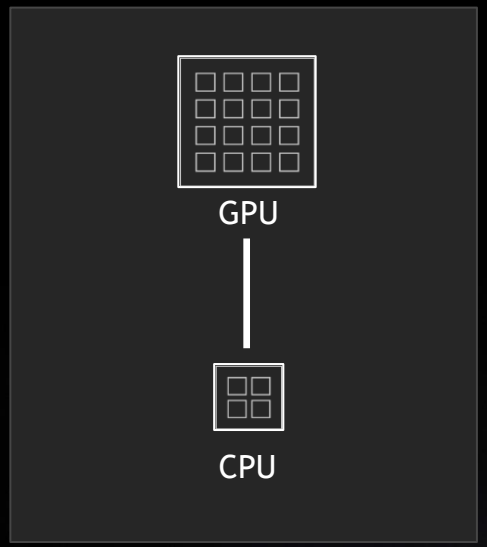


ACCELERATED COMPUTING

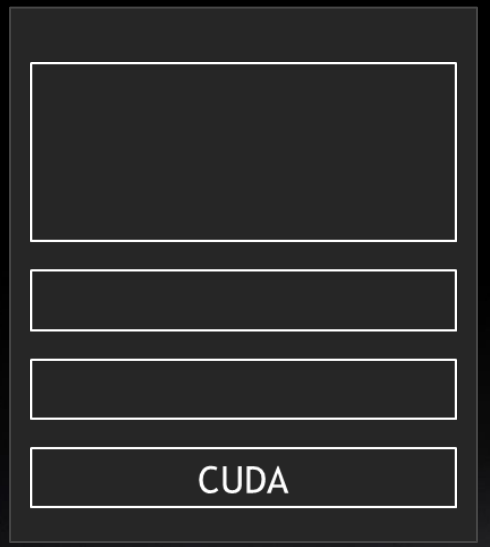


COMPUTERS WRITING SOFTWARE

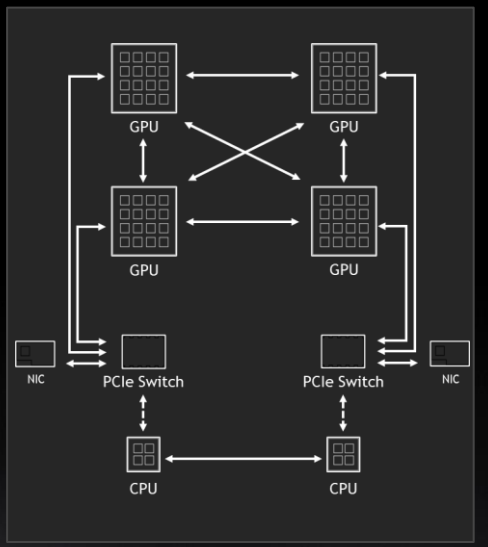
25 YEARS OF ACCELERATED COMPUTING



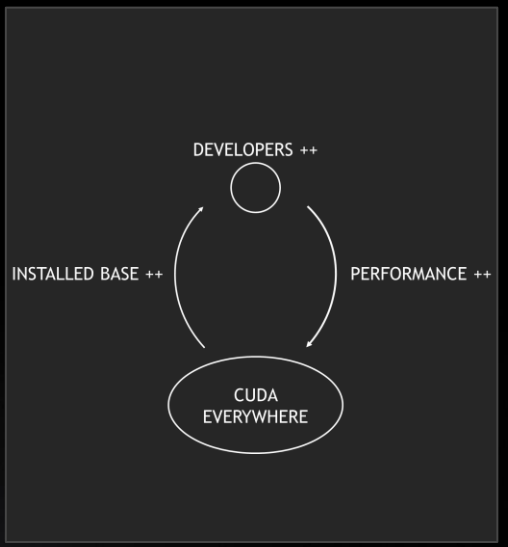
X-FACTOR SPEED UP



FULL STACK

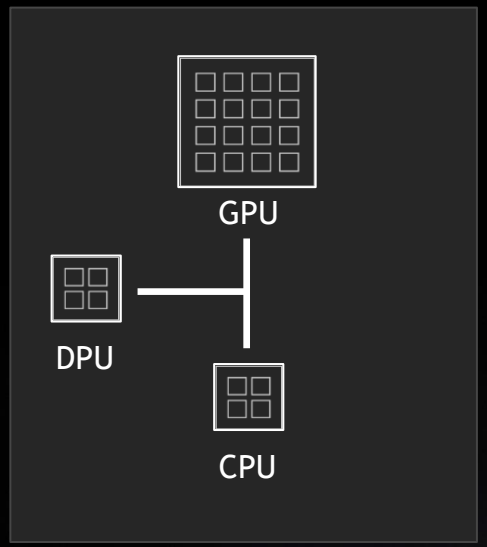


SYSTEMS

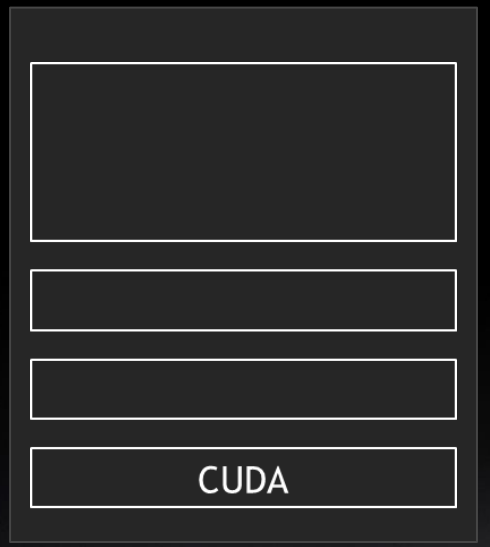


ONE ARCHITECTURE

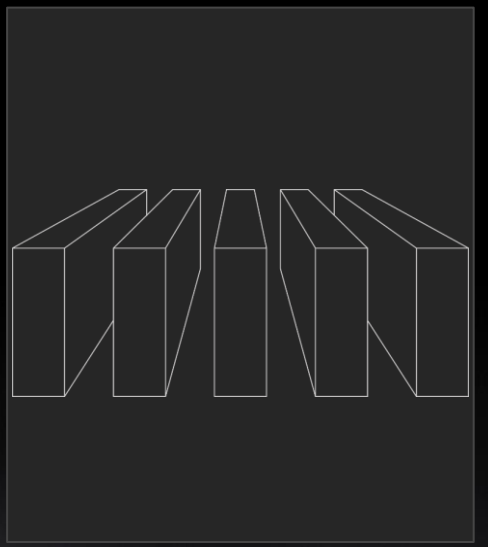
25 YEARS OF ACCELERATED COMPUTING



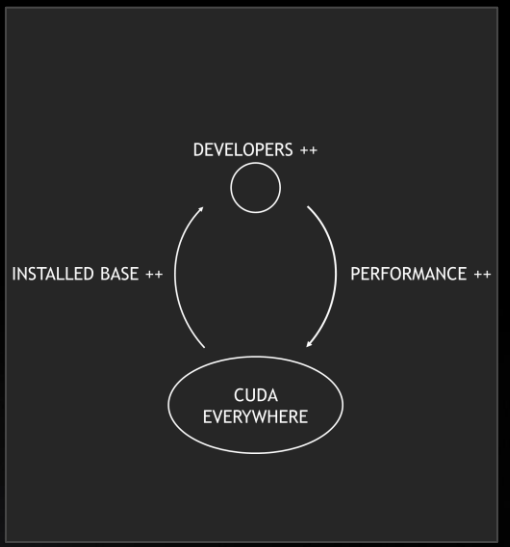
X-FACTOR SPEED UP



FULL STACK



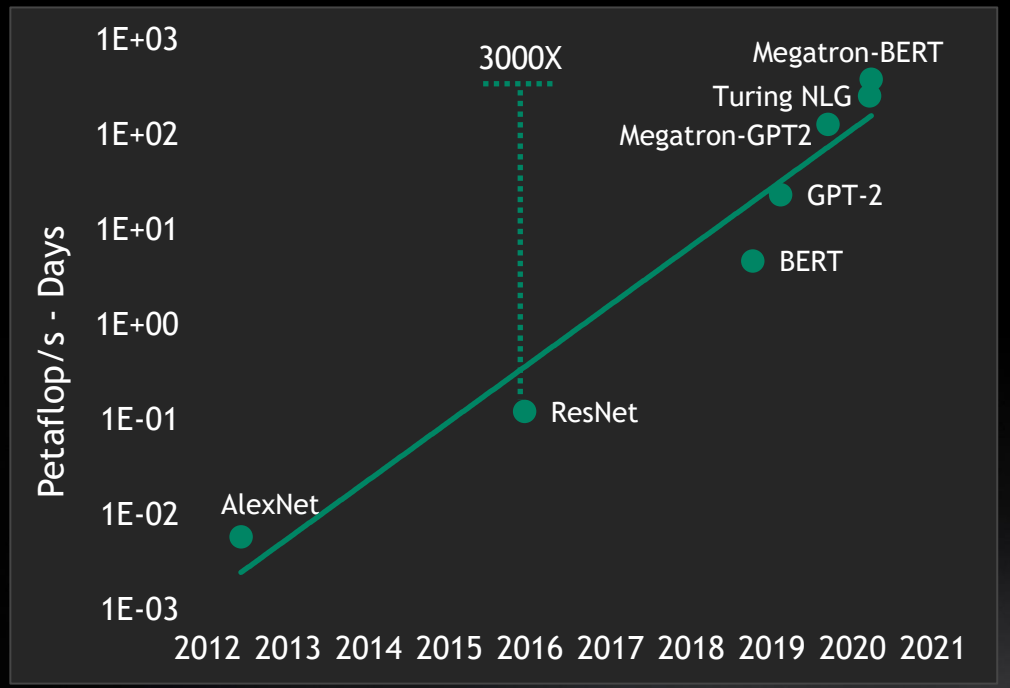
DATA-CENTER SCALE



ONE ARCHITECTURE

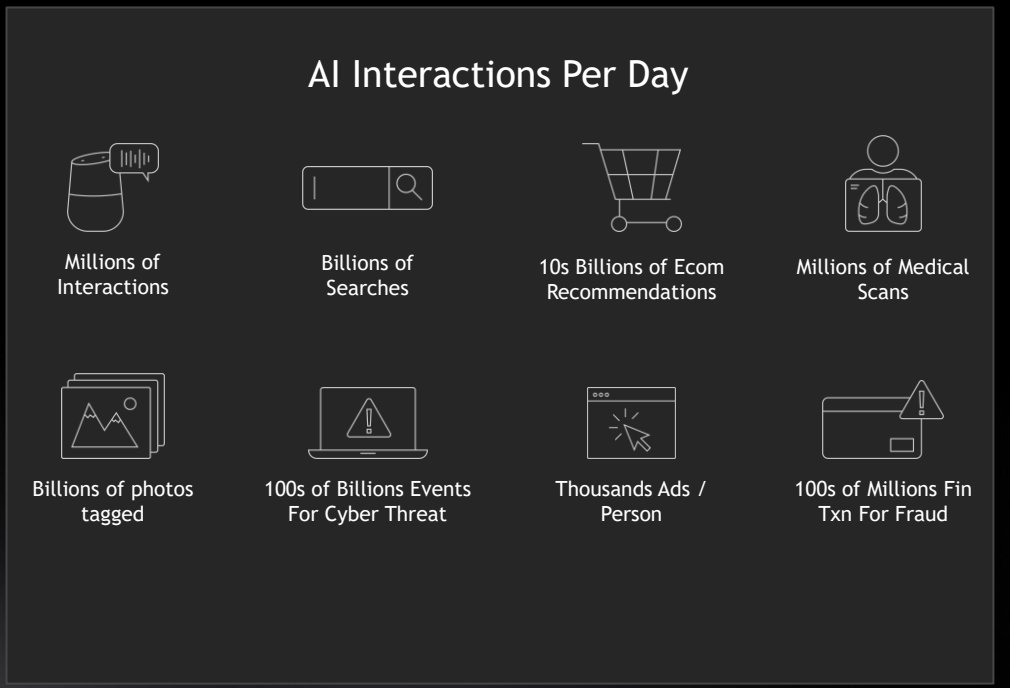
CHALLENGES: ACCELERATING BIG AND SMALL

AI Advances Demand Exponentially Higher Compute



3000X Higher Compute Required to Train Largest Models Since Volta

AI Applications Demand Distributed Pervasive Acceleration



Every AI Powered Interaction Needs Varying Amount of Compute

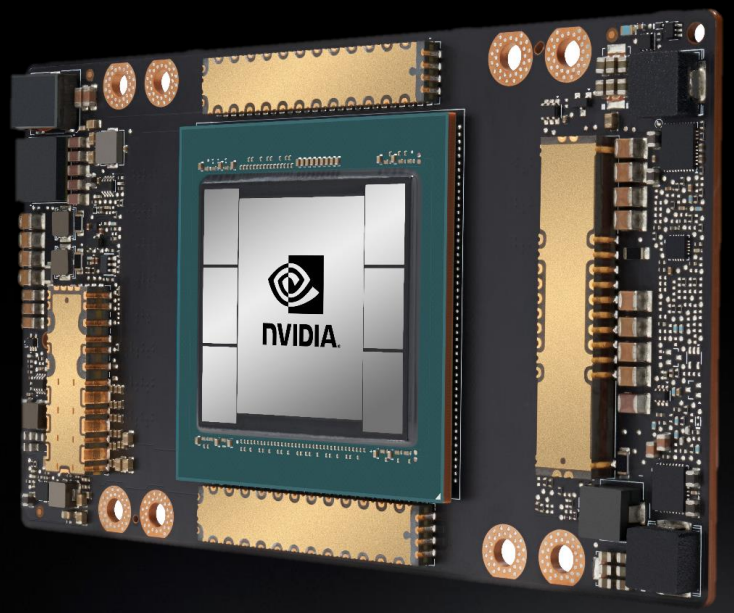
Source: OpenAI, NVIDIA

TODAY'S HYPERCONVERGED DATA CENTER

Impossible to Optimally Design Server Mix for Unpredictable Demand

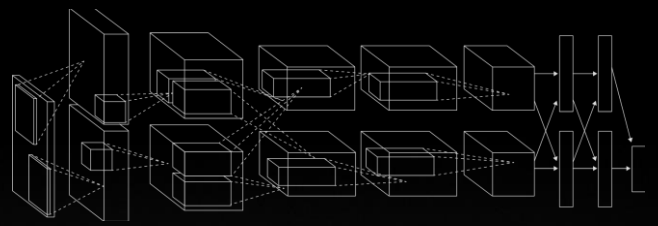
REIMAGINING THE GPU

Three Breakthroughs to Fuel the Next Era of Modern Accelerated Data Centers



20X

A GIANT LEAP IN PERFORMANCE



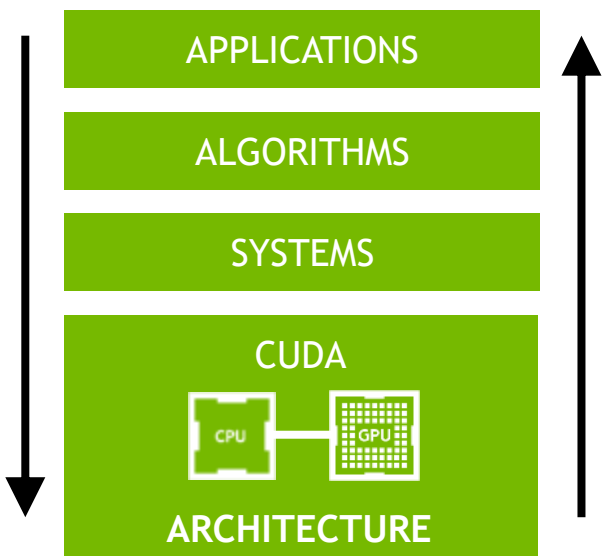
UNIFIED AI TRAINING AND INFERENCE ACCELERATION

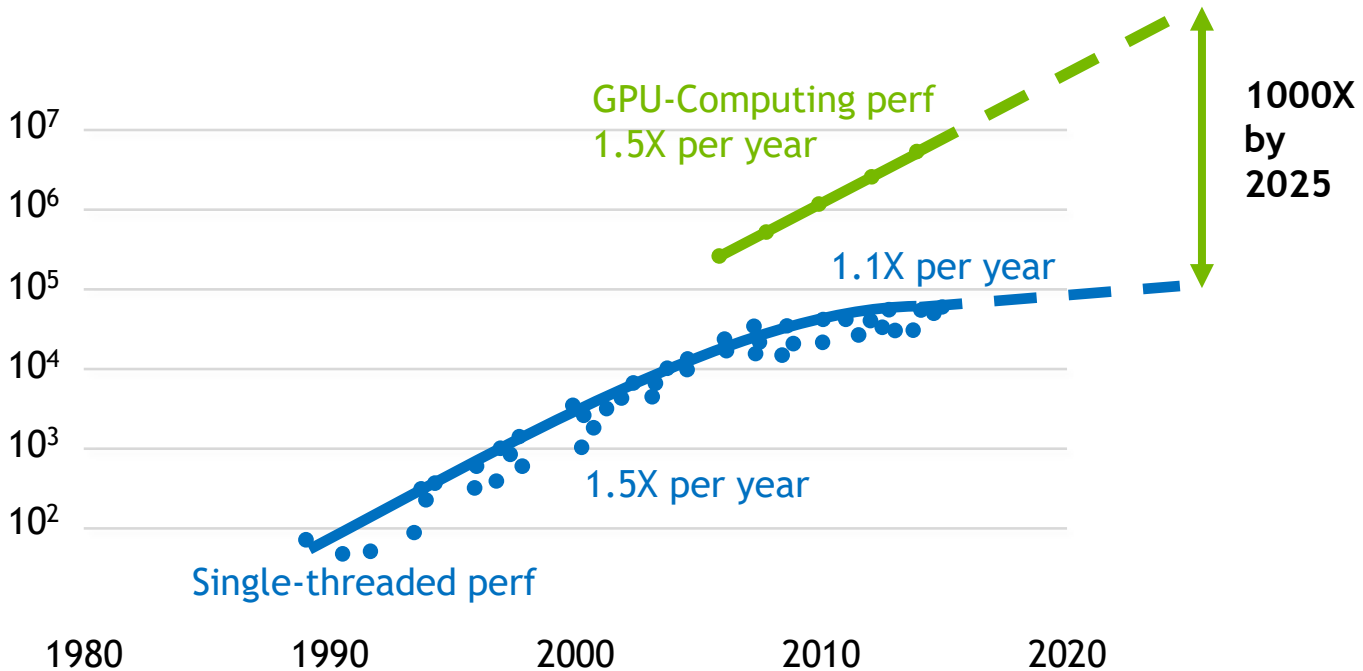
1-50

SCALABILITY FOR THE ELASTIC DATACENTER


HIGH-PERFORMANCE
COMPUTING WITH NVIDIA

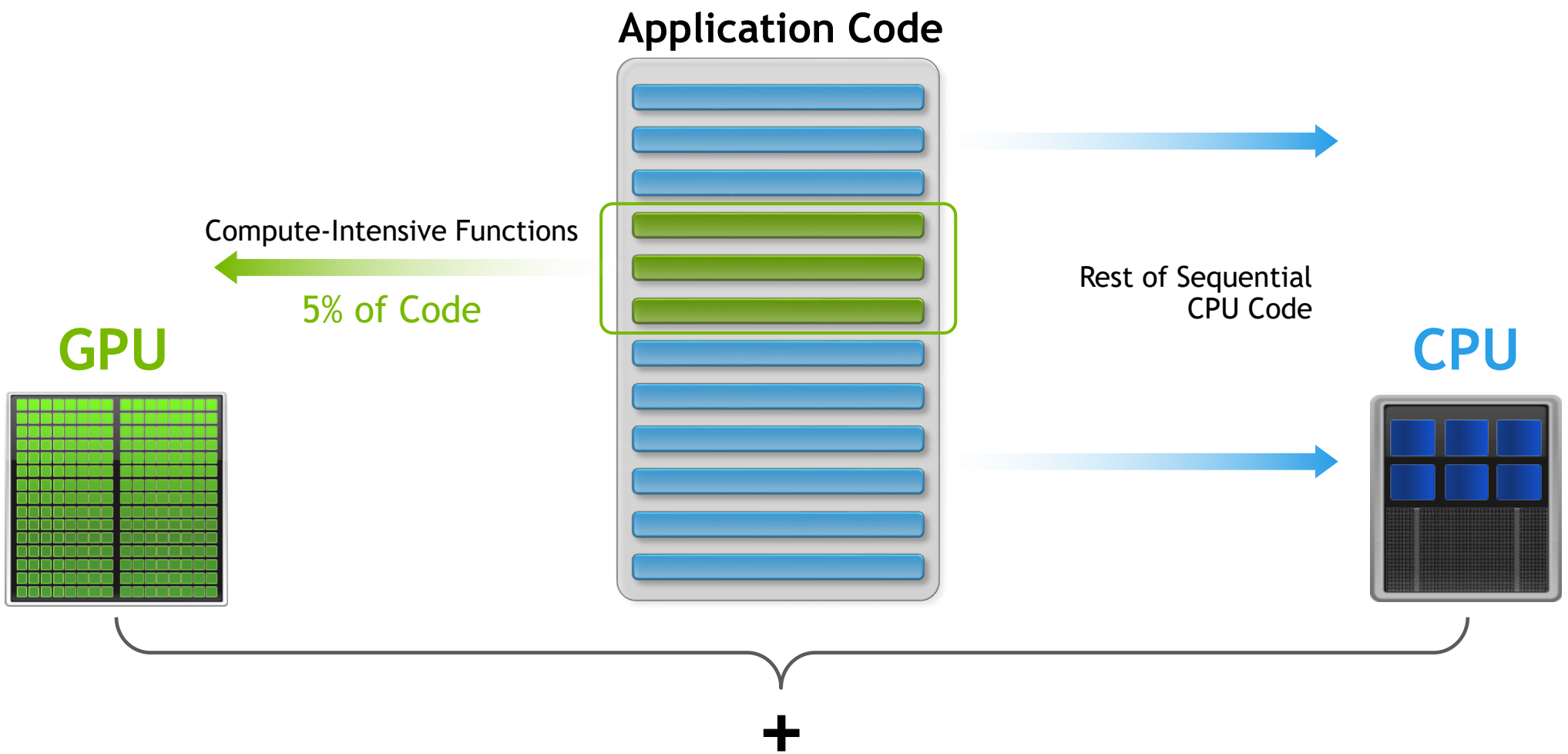
RISE OF GPU COMPUTING





Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten New plot and data collected for 2010-2015 by K. Rupp

HOW GPU ACCELERATION WORKS



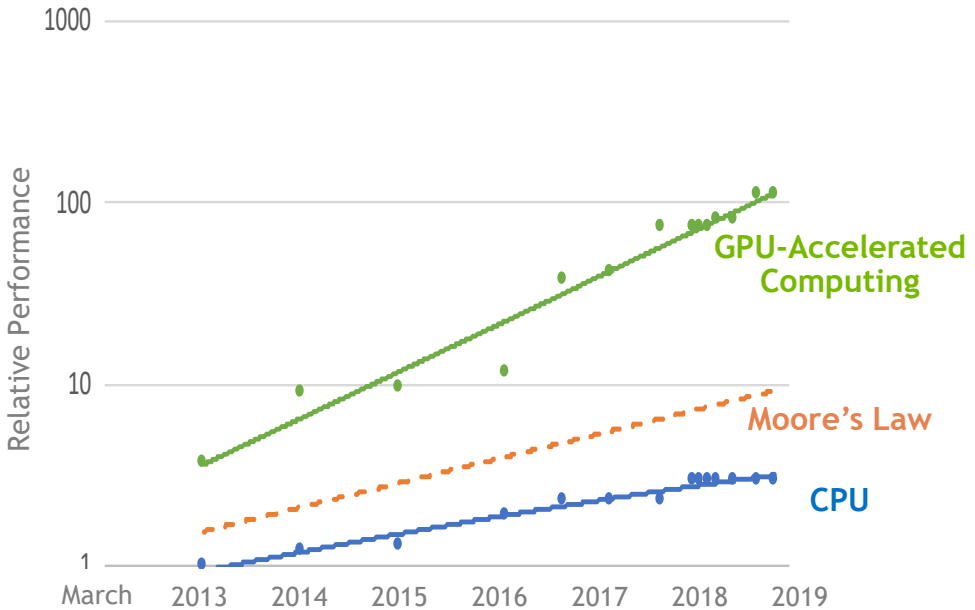
BEYOND MOORE'S LAW

Progress Of Stack In 6 Years

2013

- cuBLAS: 5.0
- cuFFT: 5.0
- cuRAND: 5.0
- cuSPARSE: 5.0
- NPP: 5.0
- Thrust: 1.5.3
- CUDA: 5.0
- Resource Mgr: r304
- Base OS: CentOS 6.2

Accelerated Server With Fermi



Measured performance of Amber, CHROMA, GTC, LAMMPS, MILC, NAMD, Quantum Espresso, SPECFEM3D

2019

- cuBLAS: 10.0
- cuFFT: 10.0
- cuRAND: 10.0
- cuSOLVER: 10.0
- cuSPARSE: 10.0
- NPP: 10.0
- Thrust: 1.9.0
- CUDA: 10.0
- Resource Mgr: r384
- Base OS: Ubuntu 16.04

Accelerated Server with Volta

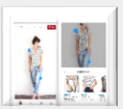
NVIDIA DATA CENTER PLATFORM

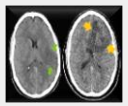
Single Platform Drives Utilization and Productivity

CUSTOMER USE CASES


Speech

Translate

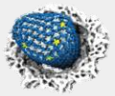

Recommender

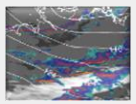

Healthcare

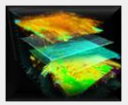

Manufacturing


Finance

CONSUMER INTERNET & INDUSTRY APPLICATIONS


Molecular Simulations


Weather Forecasting


Seismic Mapping

SCIENTIFIC APPLICATIONS


Creative & Technical


Knowledge Workers

VIRTUAL GRAPHICS

APPS & FRAMEWORKS

python™ TensorFlow mxnet Chainer ONNX RAPIDS PYTORCH

Amber NAMD +600 Applications

CATIA Ps AUTODESK 3DS MAX Windows 10

CUDA-X & NVIDIA SDKs

MACHINE LEARNING: cuDF | cuML | cuGRAPH

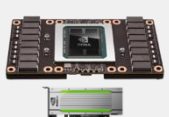
DEEP LEARNING: cuDNN | CUTLASS | TensorRT

HPC: OpenACC | cuFFT

VIRTUAL GPU: vDWS | vPC | vAPPS

CUDA & CORE LIBRARIES - cuBLAS | NCCL

TESLA GPUs & SYSTEMS


TESLA GPU


NVIDIA DGX FAMILY

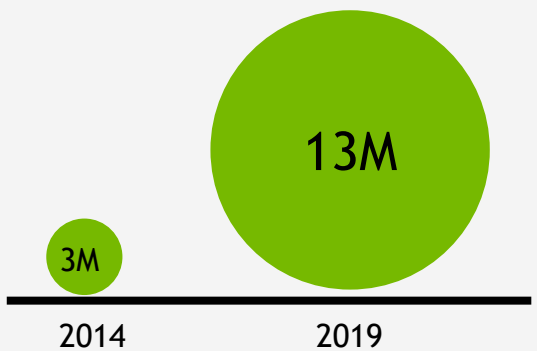

NVIDIA HGX


EVERY OEM


EVERY MAJOR CLOUD

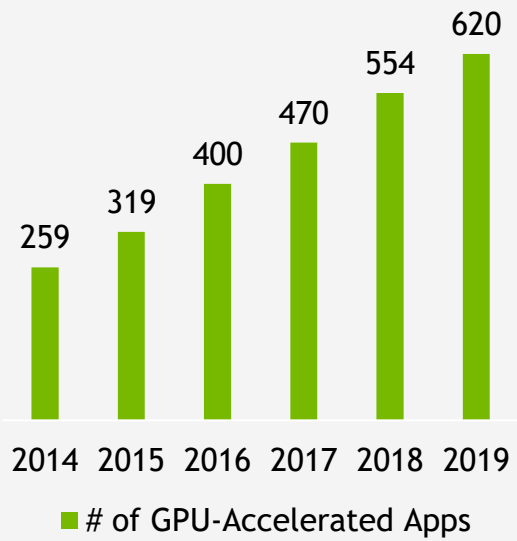
MOST ADOPTED PLATFORM FOR ACCELERATING HPC

13M CUDA Downloads



4X IN 5 YEARS

600+ Applications Accelerated



ALL TOP 15 APPLICATIONS ACCELERATED

125 Systems on Top 500

World's #1 Summit: 149 PF

World's #2 Sierra: 95 PF

Europe's #1 Piz Daint: 21 PF

Japan's #1 ABCI: 20 PF

Industrial #1 Total Pangea 3: 18 PF

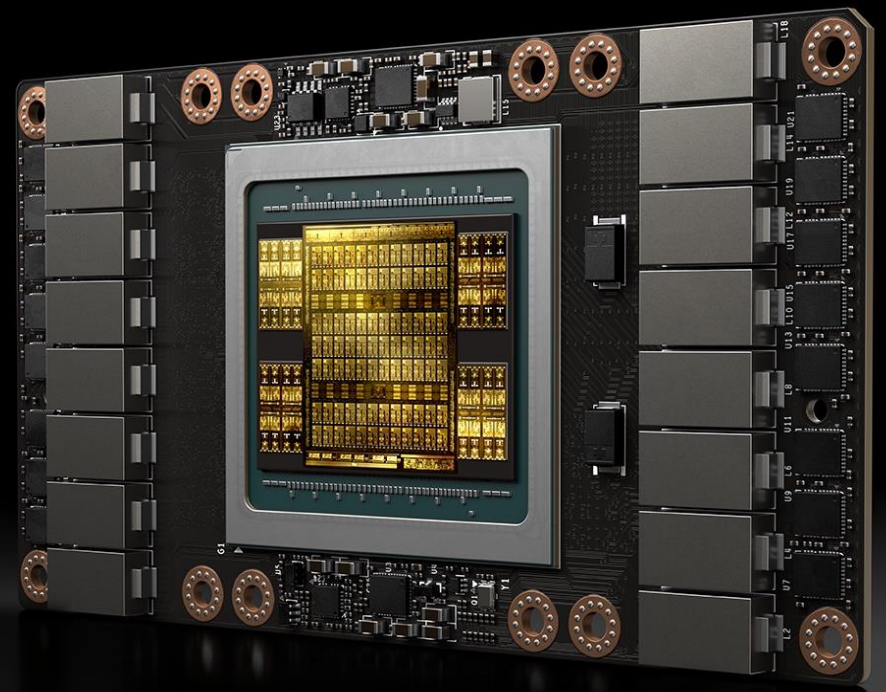
NEW HIGHS IN TOP 500 LIST

NVIDIA POWERS WORLD'S FASTEST SUPERCOMPUTER

Summit Becomes First System To Scale The 100 Petaflops Milestone

122 PF
HPC

3 EF
AI



27,648
Volta Tensor Core GPUs

NVIDIA POWERS FASTEST SUPERCOMPUTERS IN US, EUROPE, JAPAN, INDUSTRY

17 of World's 20 Most Energy-efficient Supercomputers

ORNL Summit
World's Fastest
27,648 GPUs| 122 PF

LLNL Sierra
US 2nd Fastest
17,280 GPUs| 72 PF

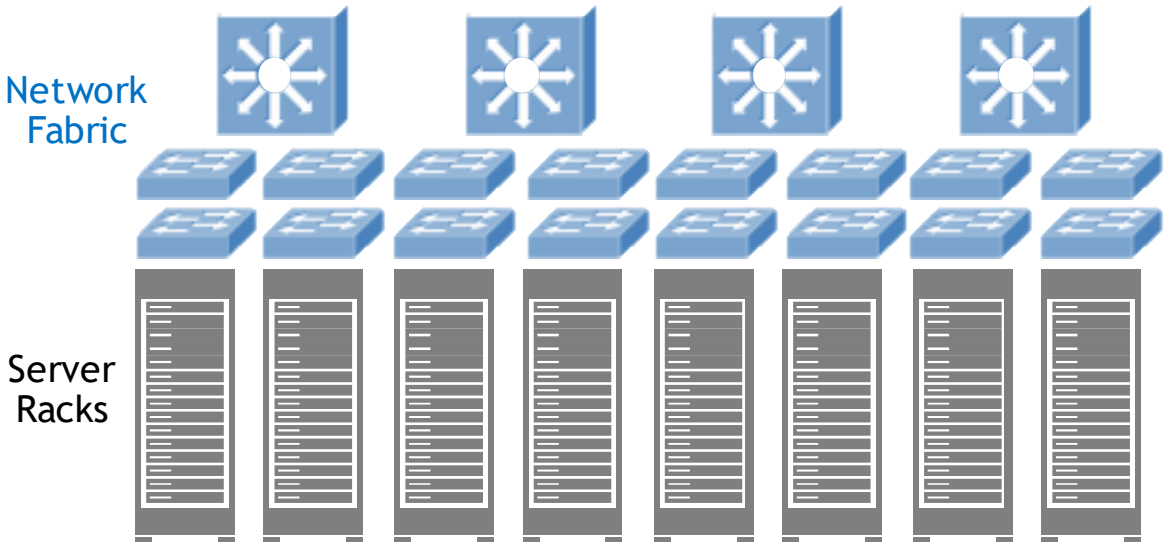
ABCI
Japan's Fastest
4,352 GPUs| 20 PF

Piz Daint
Europe's Fastest
5,320 GPUs| 20 PF

ENI HPC4
Fastest Industrial
3,200 GPUs| 12 PF

WEAK NODES

Lots of Nodes Interconnected with
Vast Network Overhead



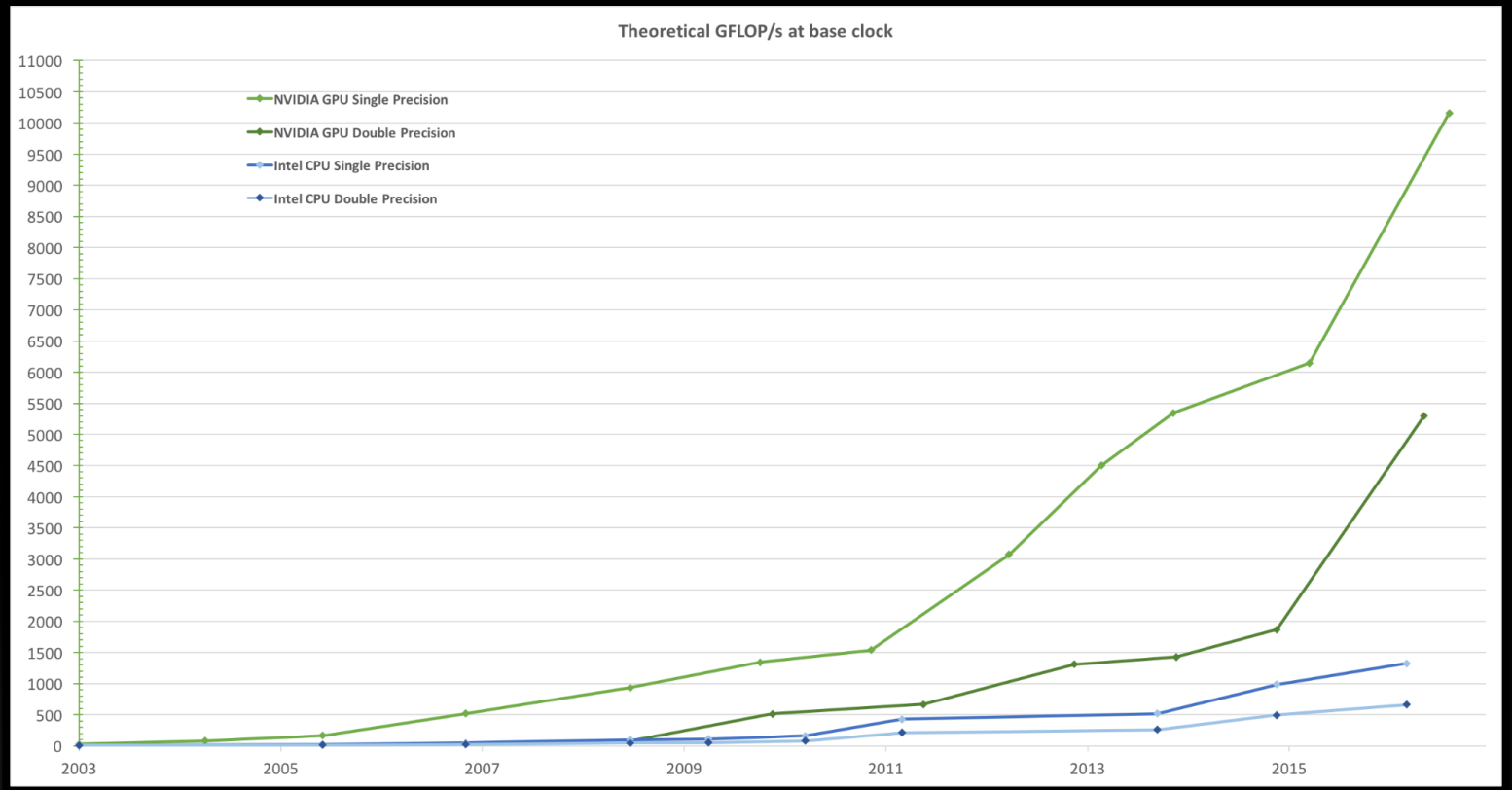
STRONG NODES

Few Lightning-Fast Nodes with
Performance of Hundreds of Weak Nodes


GPU ARCHITECTURE

SINGLE AND DOUBLE PRECISION THROUGHPUT

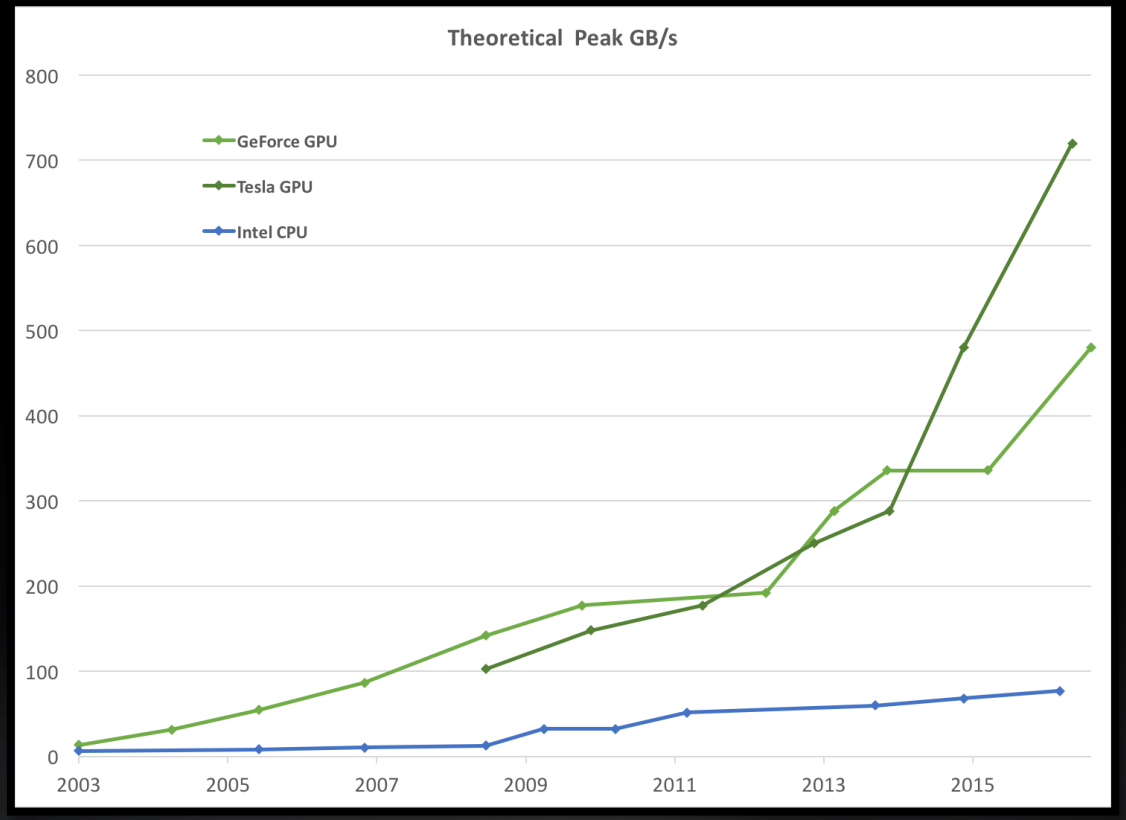
Floating-Point Operations per Second for the CPU and GPU



NVIDIA

SINGLE AND DOUBLE PRECISION THROUGHPUT

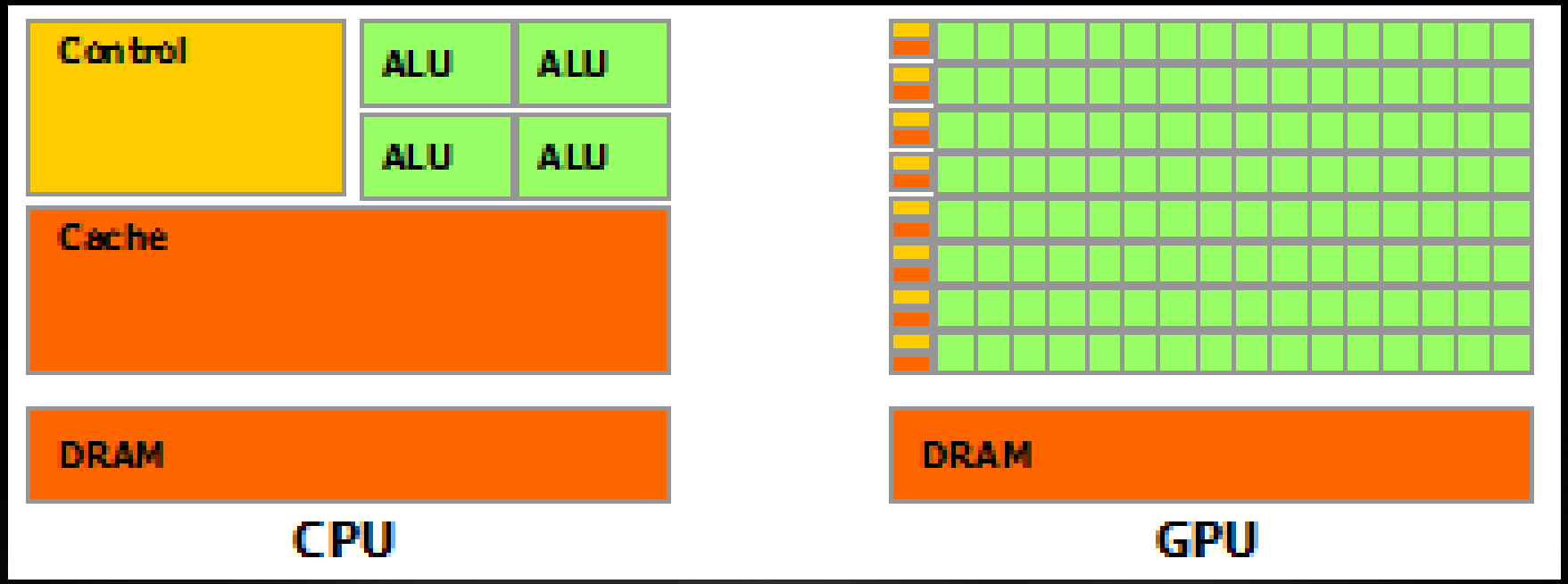
Memory Bandwidth for the CPU and GPU



Why does GPU have memory?

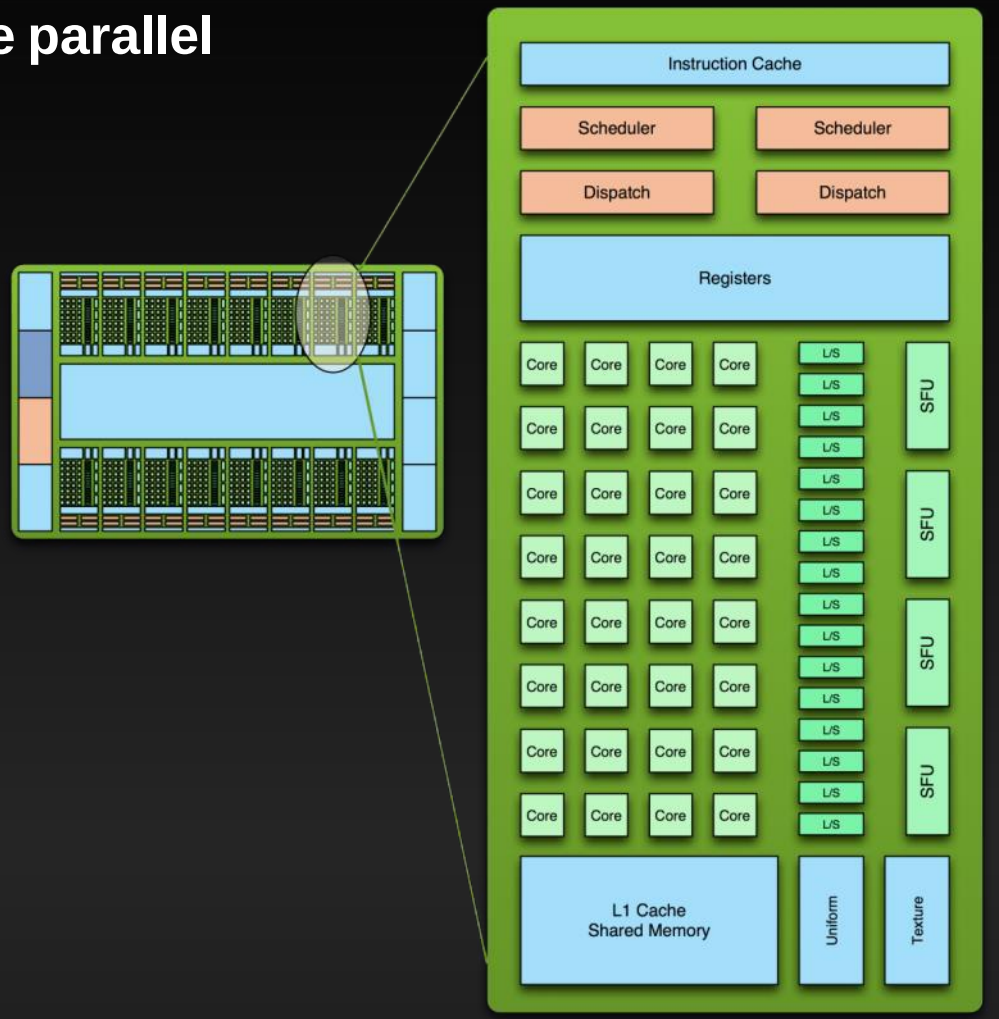
CPU VS GPU

The GPU Devotes More Transistors to Data Processing



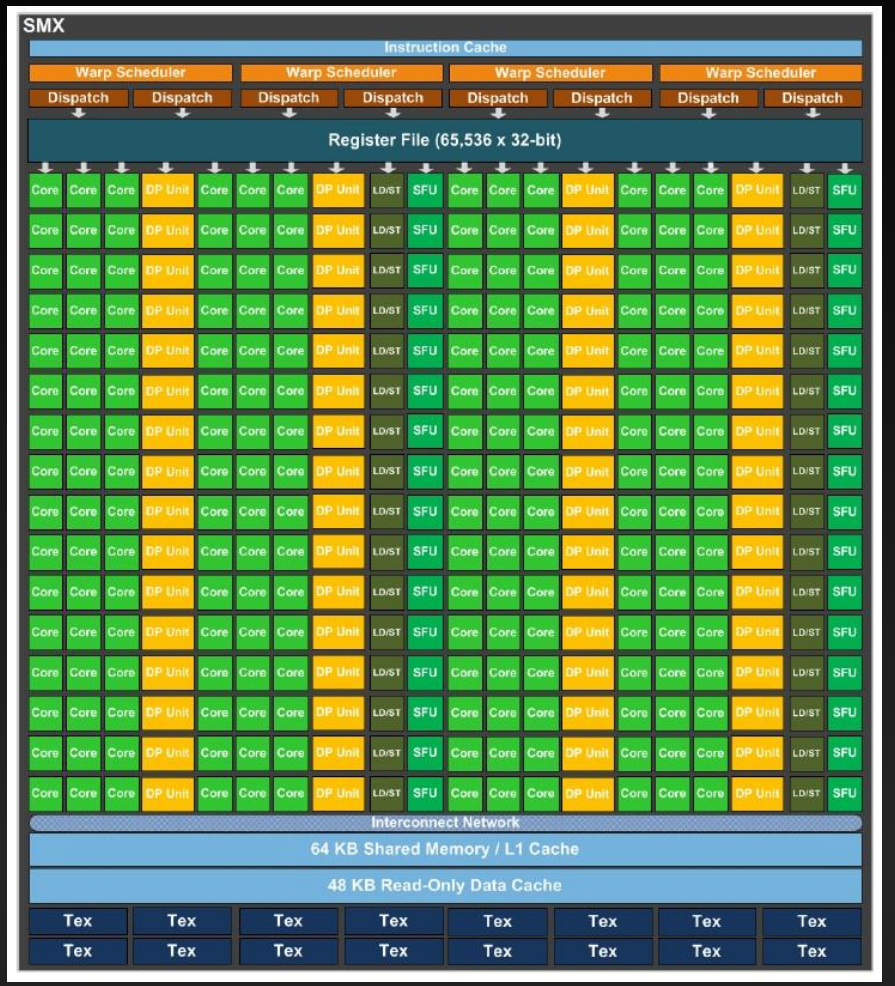
20-Series Architecture (Fermi)

- 512 *Scalar Processor (SP) cores* execute parallel thread instructions
- 16 Streaming *Multiprocessors (SMs)* each contains
  - 32 scalar processors
    - 32 fp32 / int32 ops / clock,
    - 16 fp64 ops / clock
  - 4 Special Function Units (SFUs)
  - Shared register file (128KB)
  - *48 KB / 16 KB Shared memory*
  - 16KB / 48 KB L1 data cache



Kepler cc 3.5 SM (GK110)

- “SMX” (enhanced SM)
- 192 SP units (“cores”)
- 64 DP units
- LD/ST units, 64K registers
- 4 warp schedulers
- Each warp scheduler is dual-issue capable
- K20: 13 SMX’s, 5GB
- K20X: 14 SMX’s, 6GB
- K40: 15 SMX’s, 12GB



Maxwell/Pascal cc5.2, cc6.1 SM

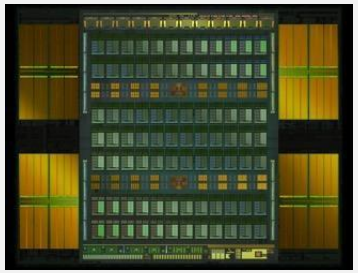
- "SMM" (enhanced SM)
- 128 SP units ("cores")
- 4 DP units
- LD/ST units
- cc 6.1: INT8
- 4 warp schedulers
- Each warp scheduler is dual-issue capable
- M40: 24 SMM's, 12/24GB
- P40: 30 SM's, 24GB
- P4: 20 SM's, 8GB

Pascal/Volta cc6.0/7.0

- 64 SP units (“cores”)
- 32 DP units
- LD/ST units
- FP16 @ 2x SP rate
- cc7.0: TensorCore
- 4 warp schedulers
- Each warp scheduler is dual-issue capable
- P100: 50 SM’s, 16GB
- V100: 80 SM’s, 16/32GB

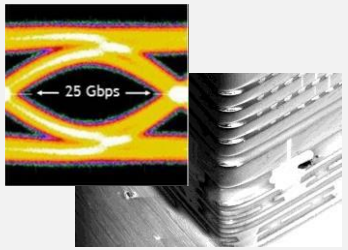
INTRODUCING TESLA V100

Volta Architecture



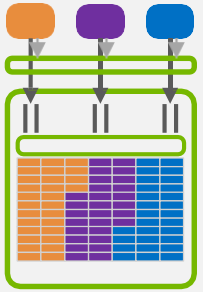
Most Productive GPU

Improved NVLink & HBM2



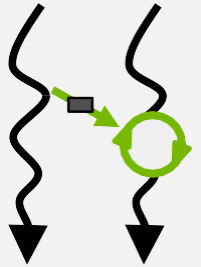
Efficient Bandwidth

Volta MPS



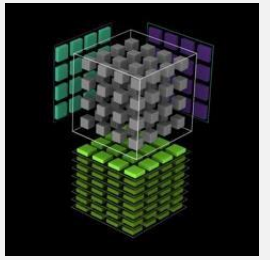
Inference Utilization

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

21B transistors
815 mm²

80 SM
5120 CUDA Cores
640 Tensor Cores

16 GB HBM2
900 GB/s HBM2
300 GB/s NVLink

*full GV100 chip contains 84 SMs

VOLTA GV100 SM

	GV100
FP32 units	64
FP64 units	32
INT32 units	64
Tensor Cores	8
Register File	256 KB
Unified L1/Shared memory	128 KB
Active Threads	2048

VOLTA GV100 SM

REDESIGNED FOR PRODUCTIVITY

Twice the schedulers

Simplified Issue Logic

Large, fast L1 cache

Improved SIMT model

Tensor acceleration

=

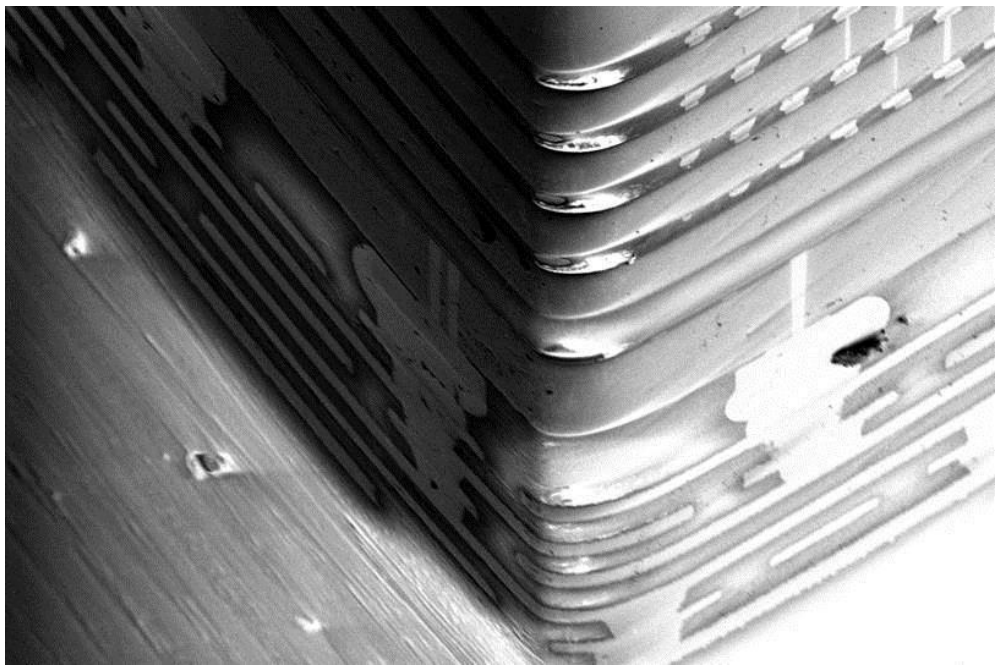
The easiest SM to program yet

GPU PERFORMANCE COMPARISON

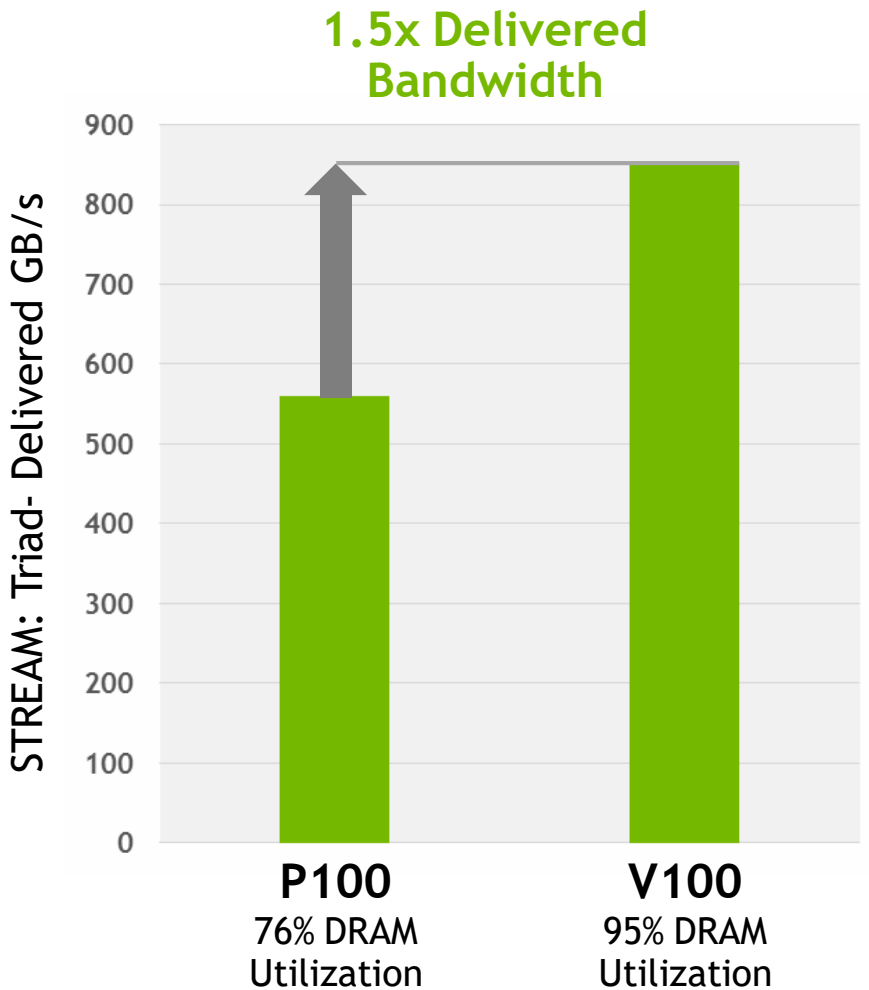
	P100	V100	Ratio
Training acceleration	10 TOPS	120 TOPS	12x
Inference acceleration	21 TFLOPS	120 TOPS	6x
FP64/FP32	5/10 TFLOPS	7.5/15 TFLOPS	1.5x
HBM2 Bandwidth	720 GB/s	900 GB/s	1.2x
NVLink Bandwidth	160 GB/s	300 GB/s	1.9x
L2 Cache	4 MB	6 MB	1.5x
L1 Caches	1.3 MB	10 MB	7.7x

MEMORY HIERARCHY

NEW HBM2 MEMORY ARCHITECTURE



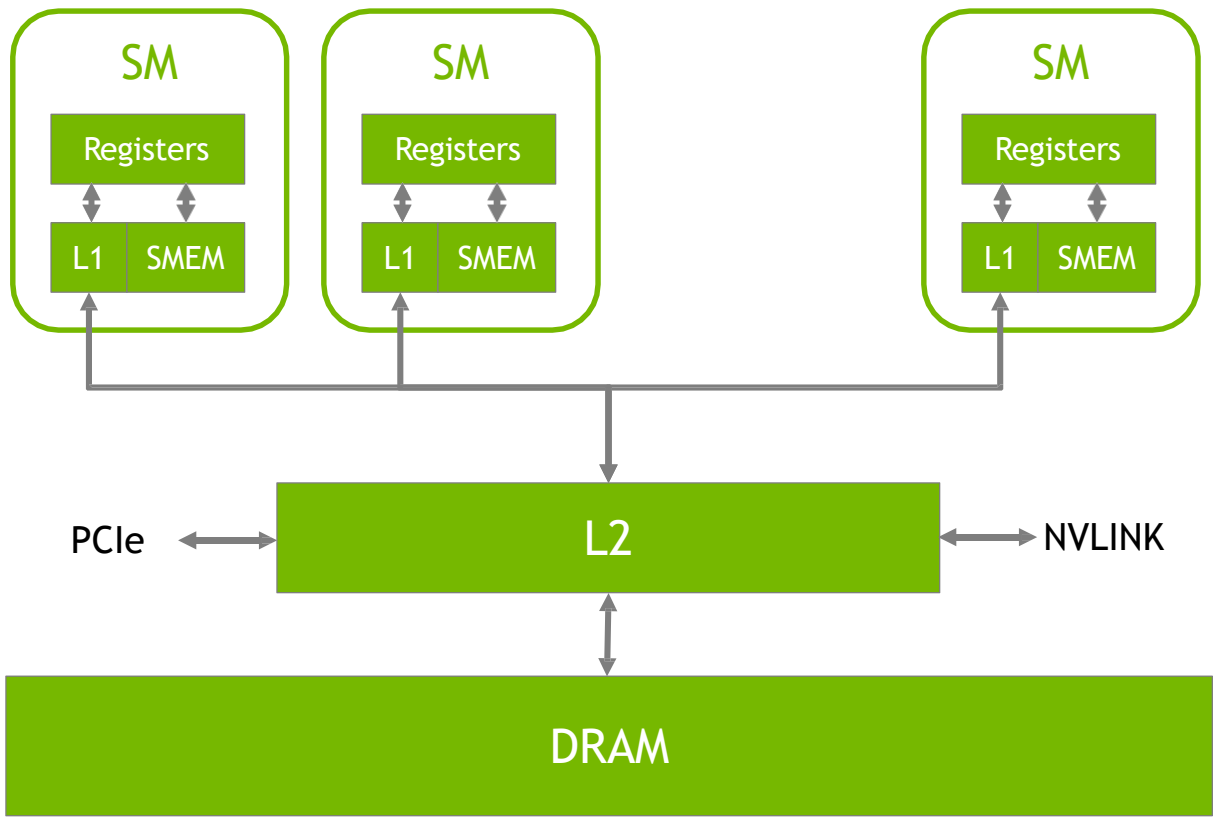
HBM2 stack



V100 measured on pre-production hardware.

VOLTA MEMORY SUBSYSTEM

Tesla V100



80 Streaming Multiprocessors
256KB register file (20 MB)

Unified Shared Mem / L1 Cache
128KB, variable split (10MB Total,
14 TB/s), Volta caches L1 writes

6 MB L2 Cache, L2 is write back

16/32 GB HBM2 (900 GB/s)

L1, L2 CACHES

Why do GPU have caches?

Caches on GPUs can help with:

"Smoothing" irregular, unaligned access patterns

Caching common data accessed by many threads

Faster register spills, local memory

Can help in codes that don't use shared memory

SHARED MEMORY

Scratch-pad memory on each SM
User-managed cache, hardware does not evict data
Data written to SMEM stays there until this the code overwrites the data or threadblock finishes execution

Useful for:
Storing frequently-accessed data, to reduce DRAM accesses
Communication among threads of a threadblock

Performance benefits compared to DRAM:
20-40x lower latency
~15x higher bandwidth

UNIFIED SHARED MEM / L1 CACHE

Variable split

Volta: 6 possible smem / L1 splits
96KB / 32KB
64KB / 64KB
32KB / 96KB
16KB / 112KB
8KB / 120KB
0KB /128 KB

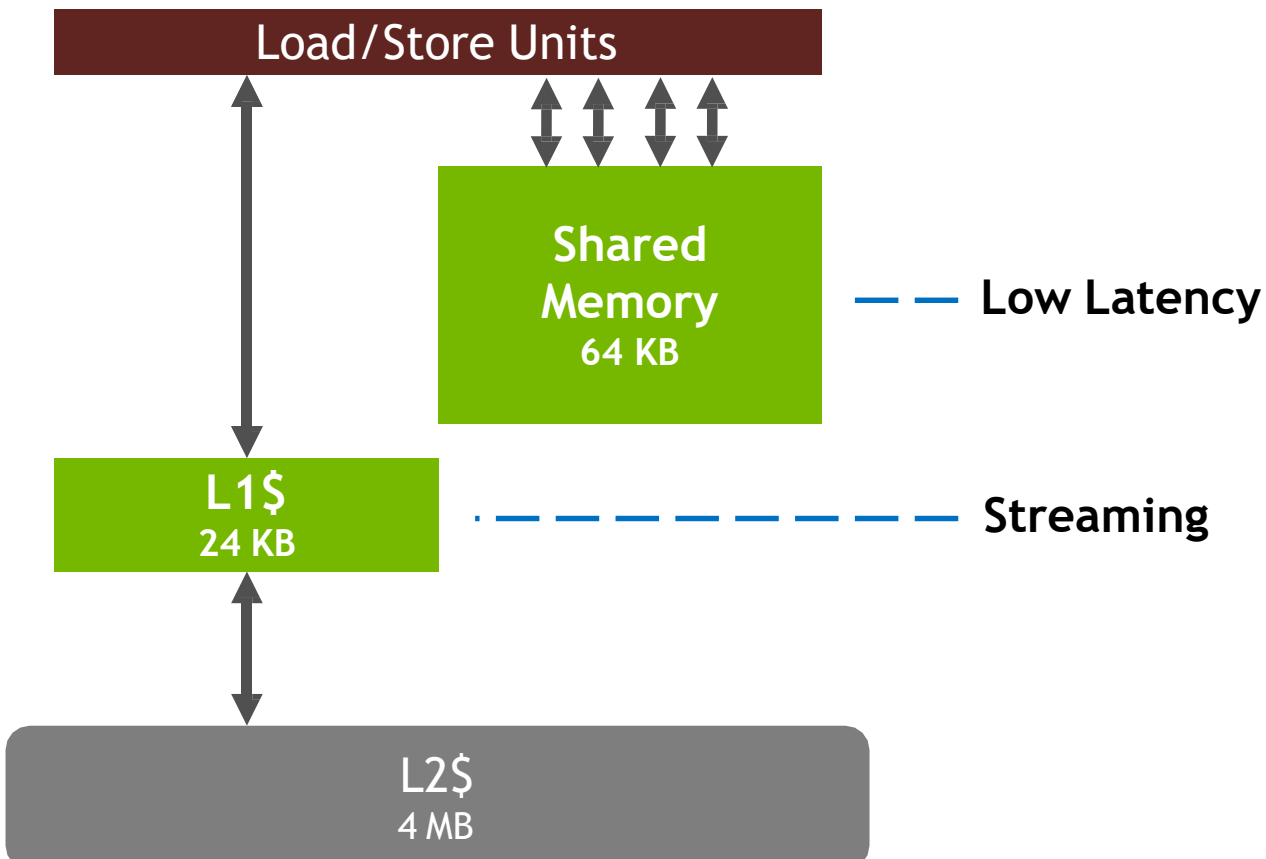
Turing: 2 possible smem / L1 splits
64KB / 32KB
32KB / 64KB

How to specify the L1 / Smem split:
cudaFuncSetAttribute (MyKernel, cudaFuncAttributePreferredSharedMemoryCarveout, carveout);

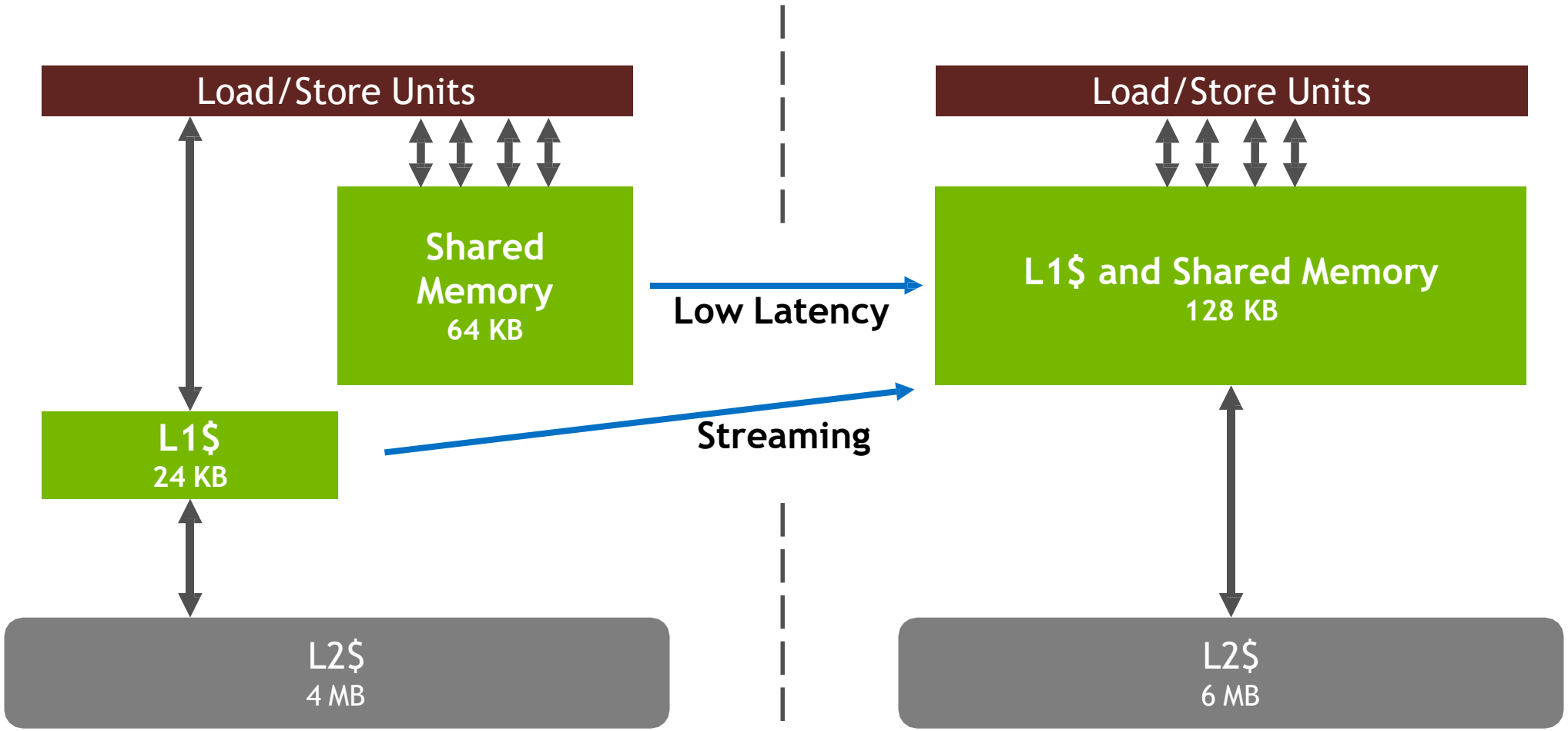
The driver usually does a pretty good job at choosing the right split.

To overcome 48 KB per threadblock limitation call:
cudaFuncSetAttribute (MyKernel, cudaFuncAttributeMaxDynamicSharedMemorySize, maxsize);

RECAP: PASCAL L1 AND SHARED MEMORY



UNIFYING KEY TECHNOLOGIES



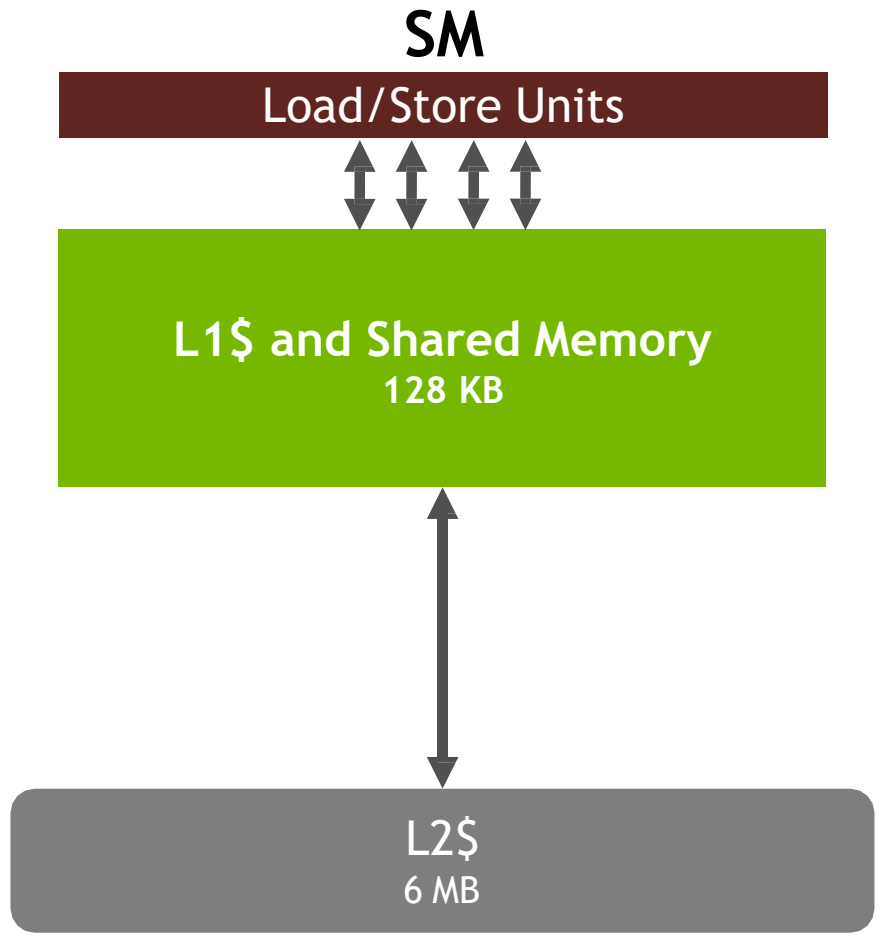
VOLTA L1 AND SHARED MEMORY

Volta Streaming L1$:

Low cache hit latency
4x more bandwidth
5x more capacity

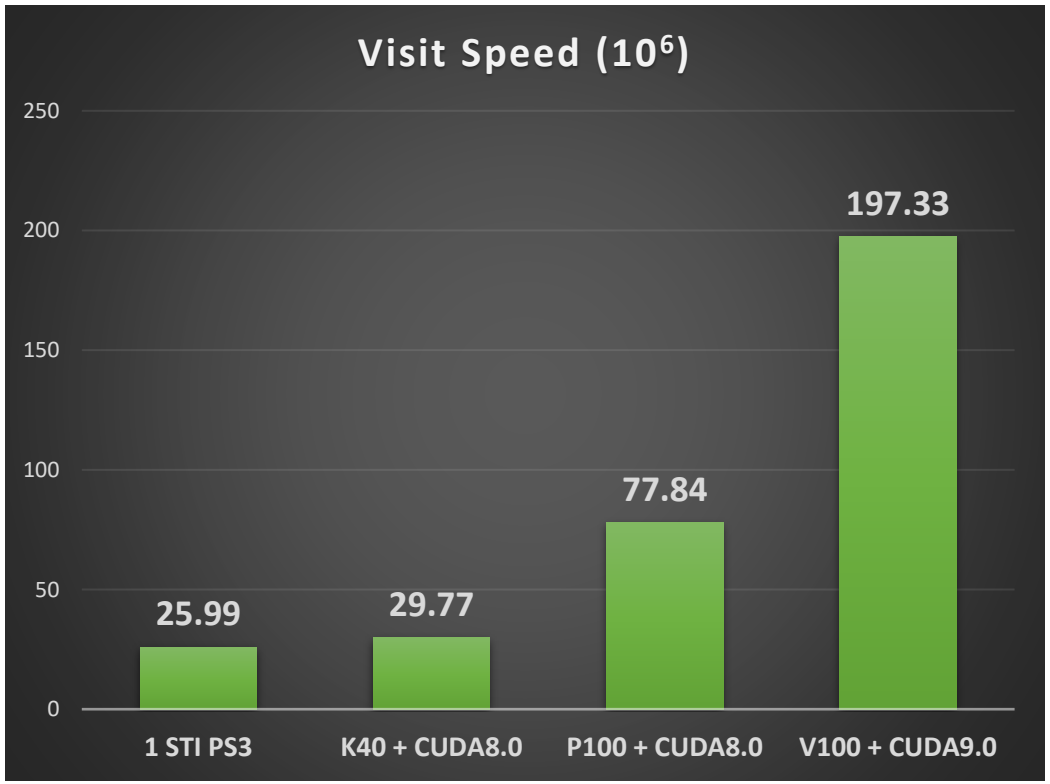
Volta Shared Memory :

Unified storage with L1
Configurable up to 96KB



“SCALABILITY OF CPU AND GPU SOLUTIONS OF THE PRIME ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM”

Jairo Panetta (ITA), Paulo Souza (ITA), Luiz Laranjeira (UnB), Carlos Teixeira Jr (UnB)



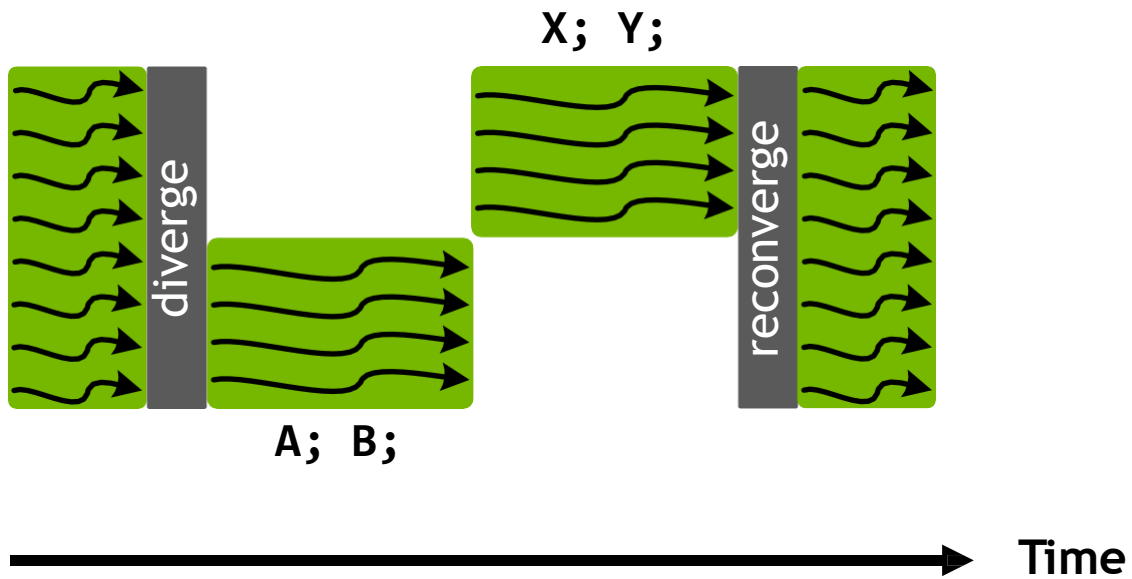
SBAC-PAD

RENASIC
Rede Nacional em Segurança da Informação e Criptografia

VOLTA WARP EXECUTION MODEL

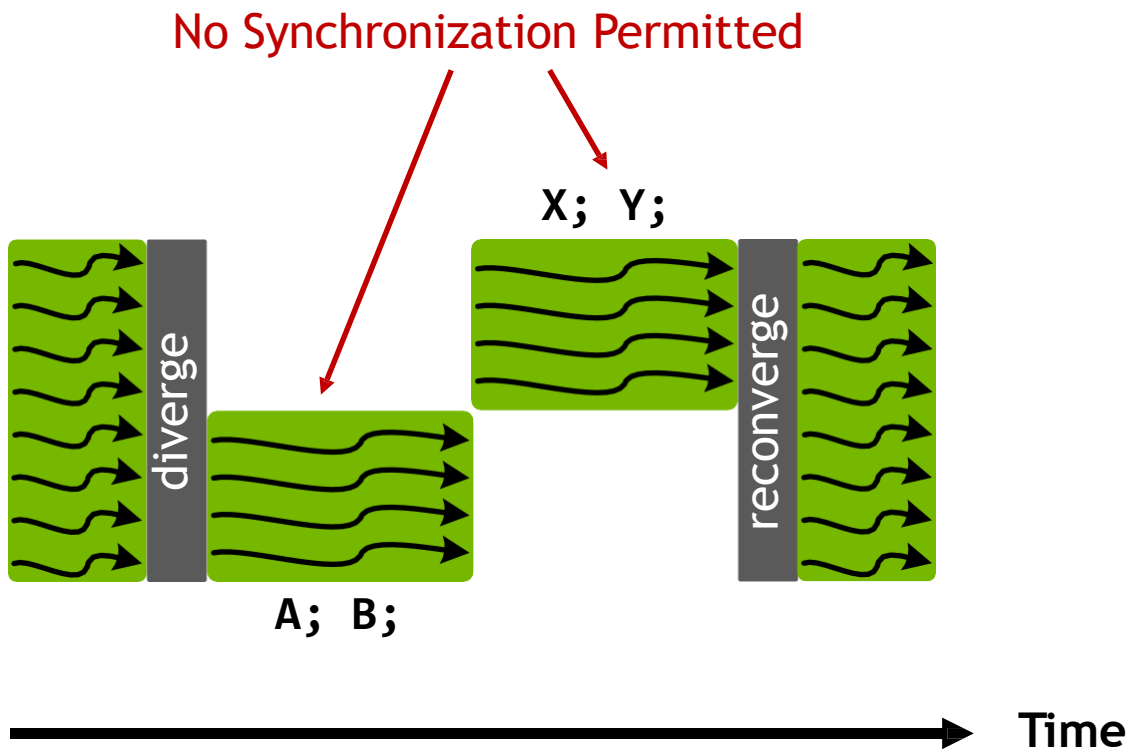
PASCAL WARP EXECUTION MODEL

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

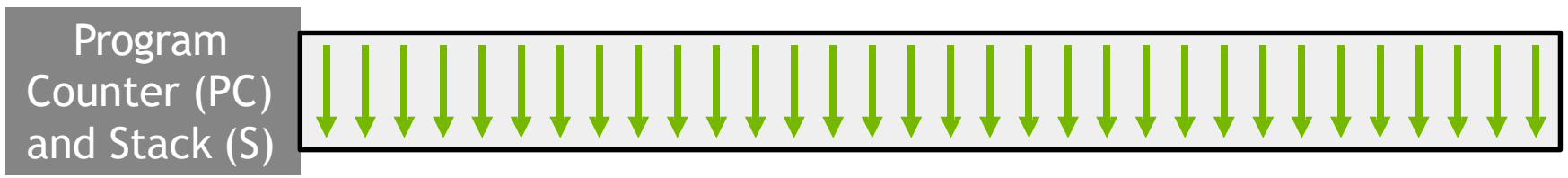


PASCAL WARP EXECUTION MODEL

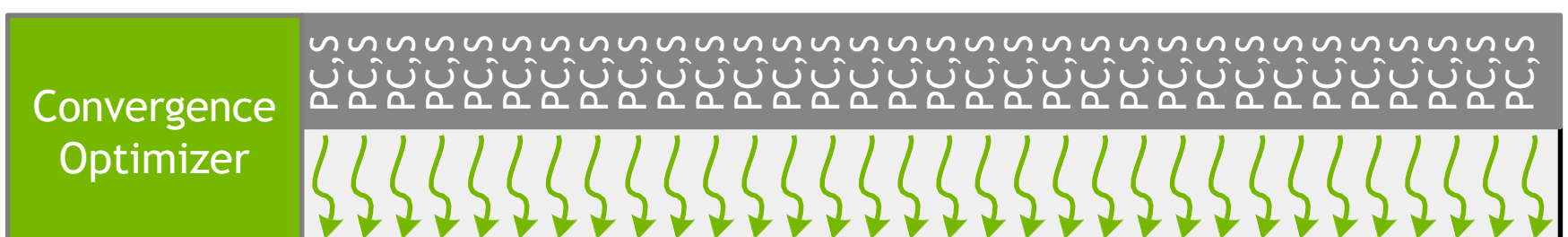
```
if (threadIdx.x < 4) {
    A;
    __syncwarp();
    B;
} else {
    X;
    __syncwarp();
    Y;
}
```



WARP IMPLEMENTATION



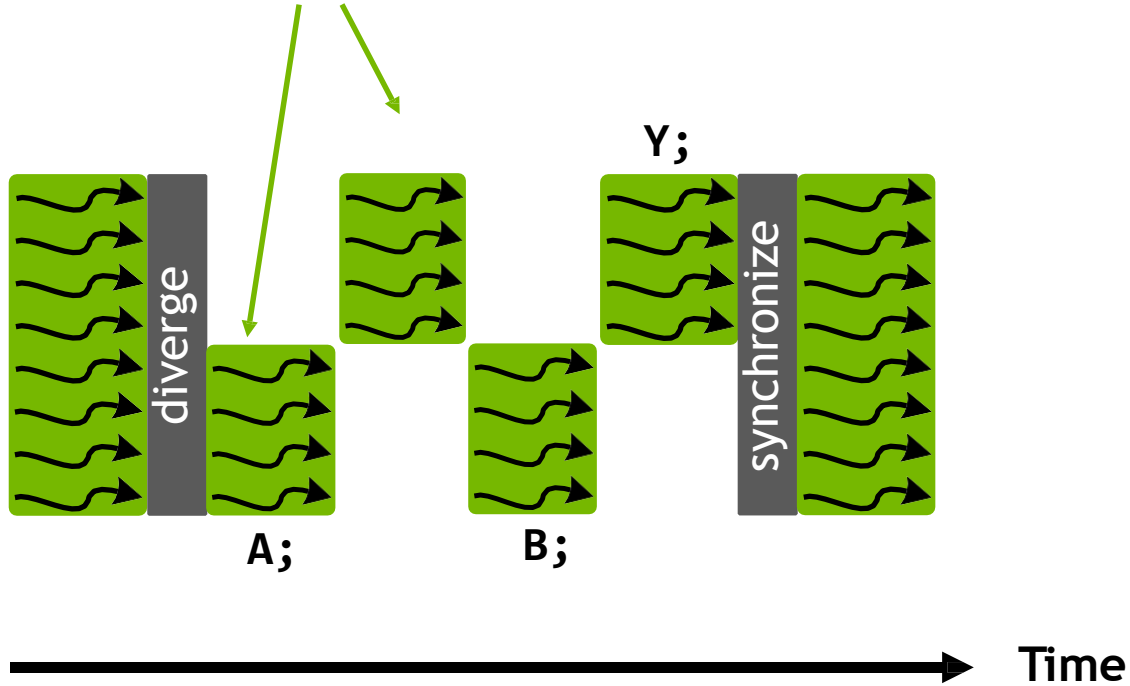
Volta



VOLTA WARP EXECUTION MODEL

Synchronization may lead to interleaved scheduling!

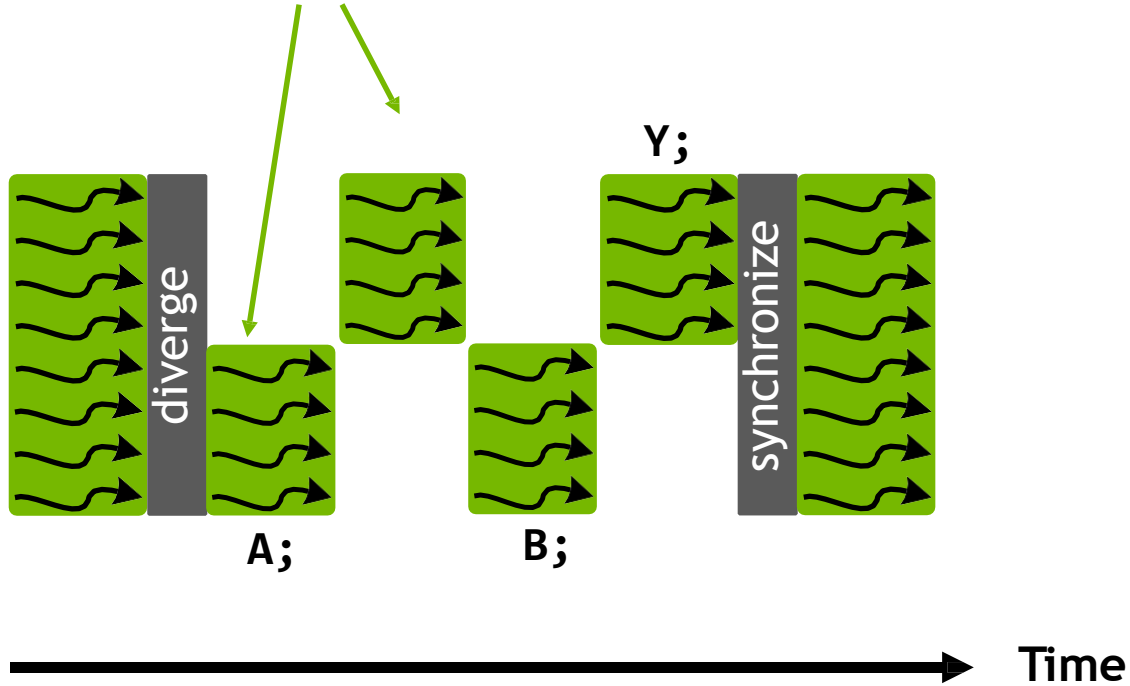
```
if (threadIdx.x < 4) {
    A;
    __syncwarp();
    B;
} else {
    X;
    __syncwarp();
    Y;
}
__syncwarp();
```



VOLTA WARP EXECUTION MODEL

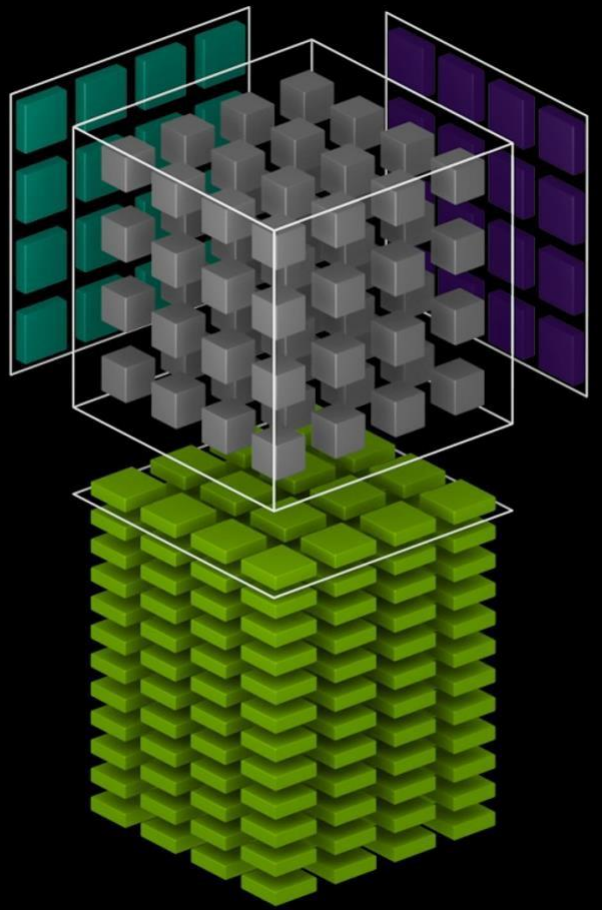
Synchronization may lead to interleaved scheduling!

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



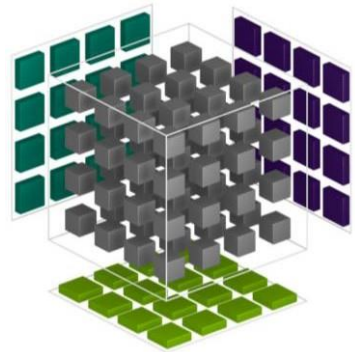
Software synchronization also supported, e.g. locks for doubly-linked list!

VOLTA TENSOR CORE



TENSOR CORE

Mixed Precision Matrix Math
4x4 matrices



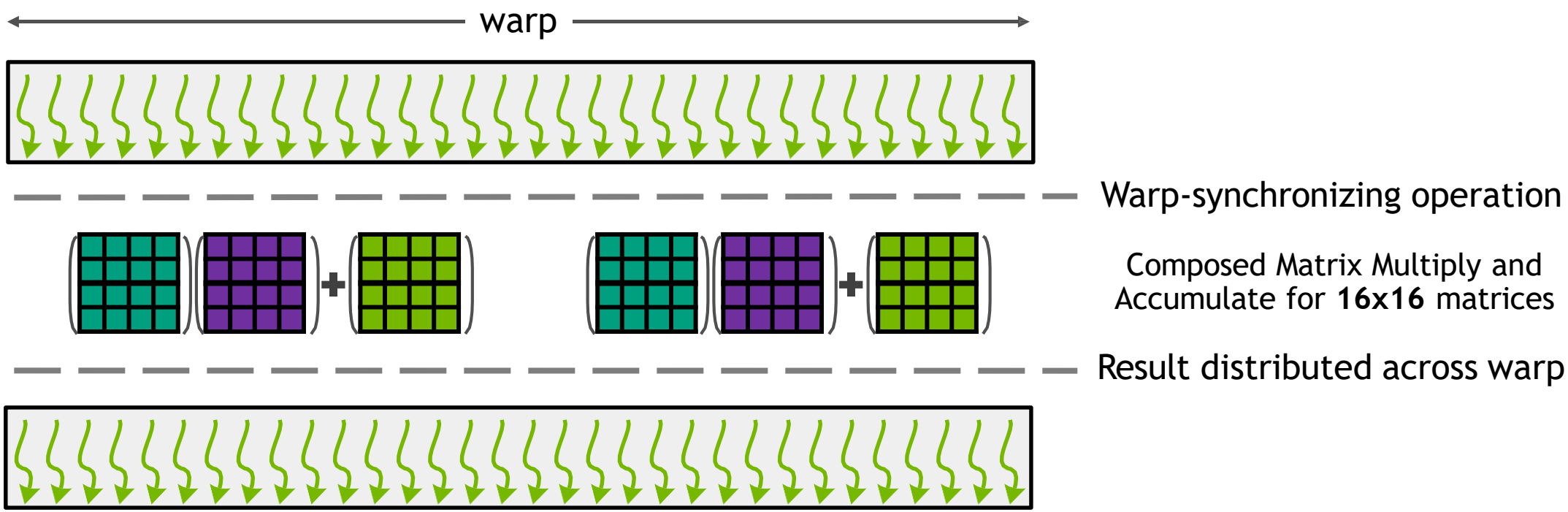
$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0}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

FP16 or FP32 | FP16 | FP16 | FP16 or FP32

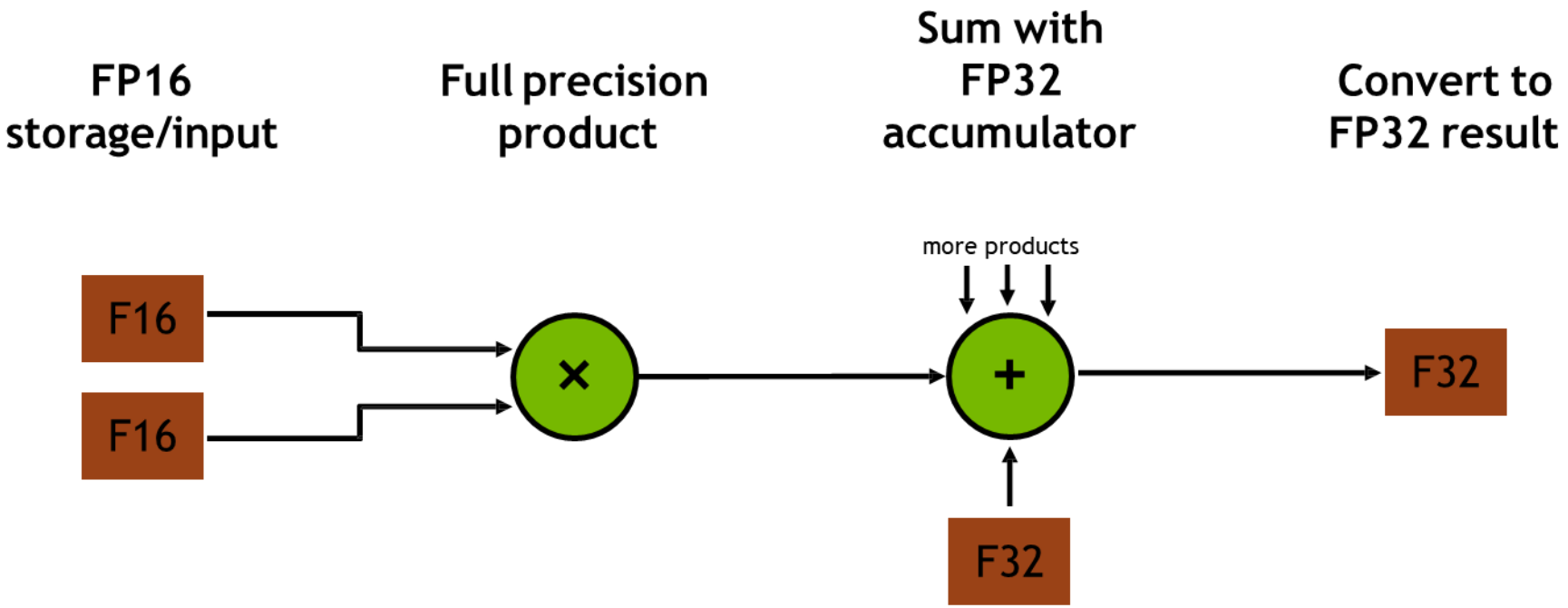
$$D = AB + C$$

TENSOR SYNCHRONIZATION

FULL WARP 16X16 MATRIX MATH

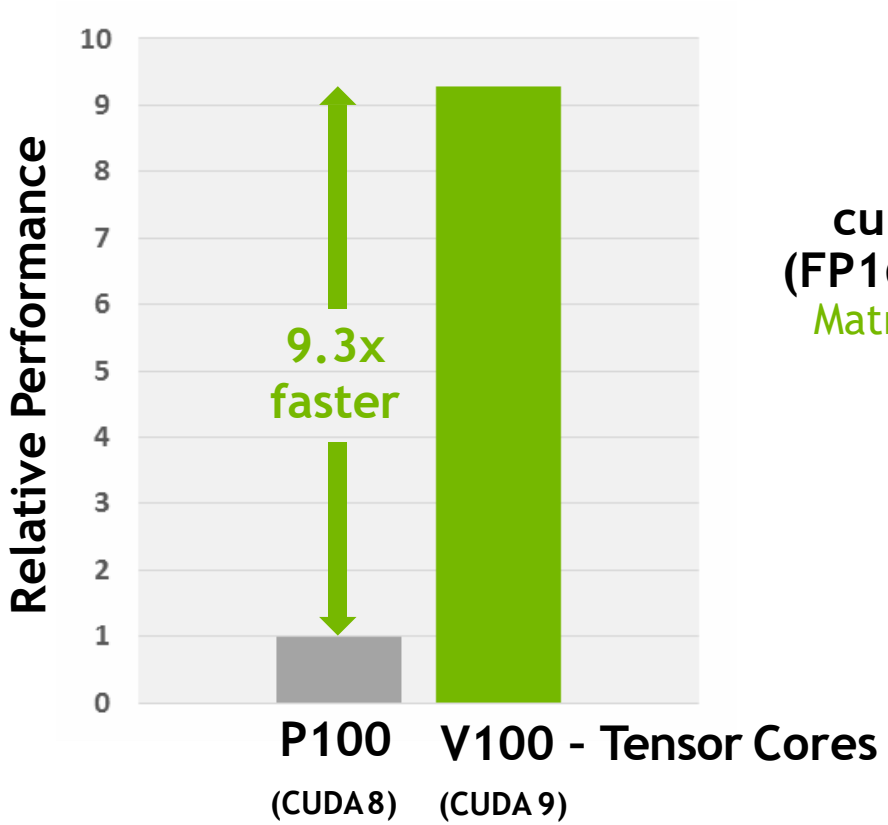


VOLTA TENSOR OPERATION



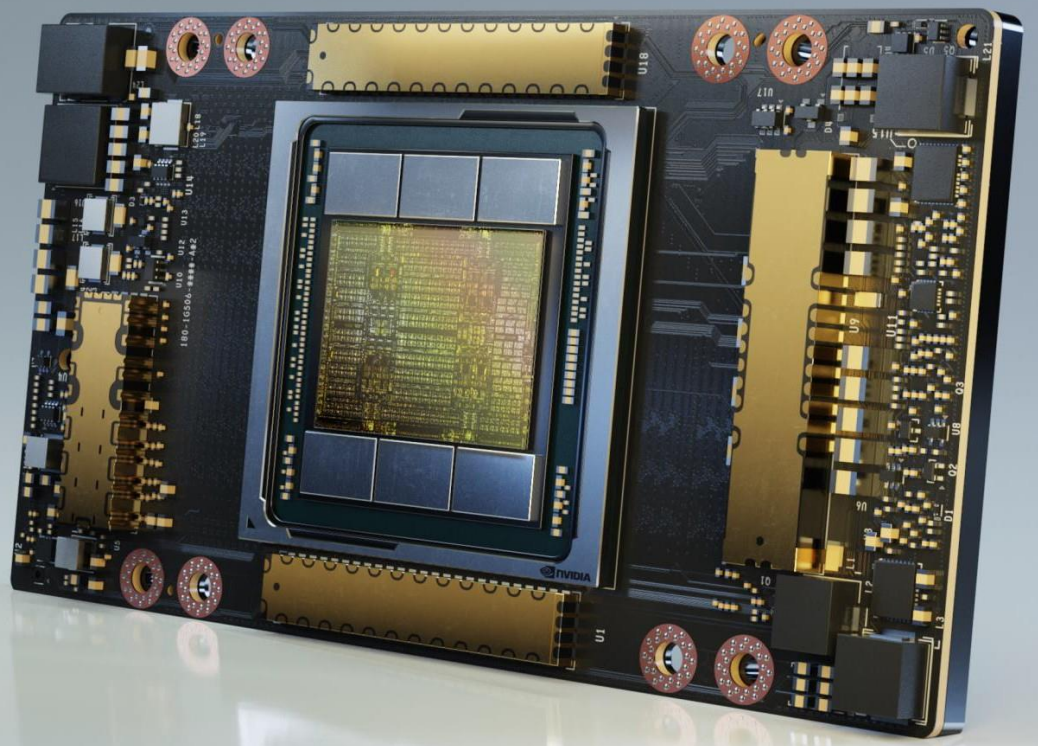
Also supports FP16 accumulator mode for inferencing

A GIANT LEAP FOR DEEP LEARNING

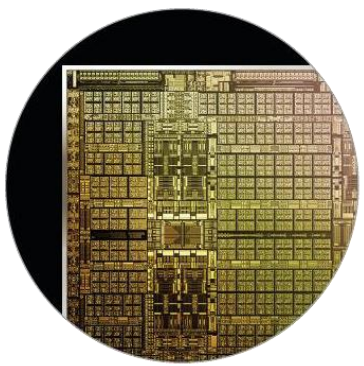


cuBLAS Mixed Precision
(FP16 input, FP32 compute)
Matrix Multiply (M=N=K=2048)

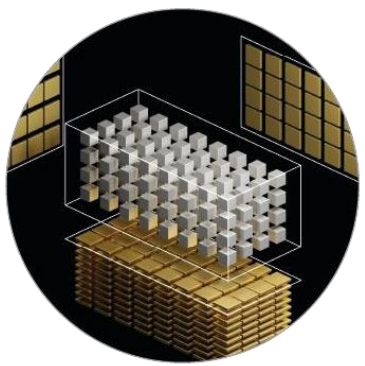
Inside the NVIDIA Ampere Architecture



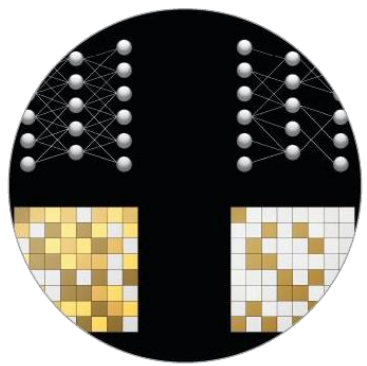
UNPRECEDENTED ACCELERATION AT EVERY SCALE



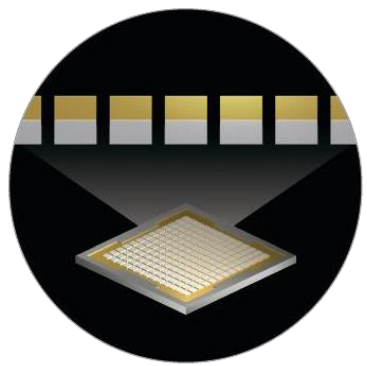
54 BILLION XTORS



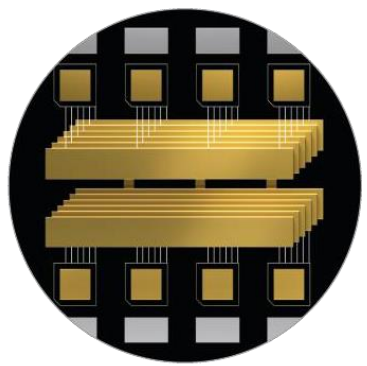
3rd GEN
TENSOR CORES



SPARSITY
ACCELERATION



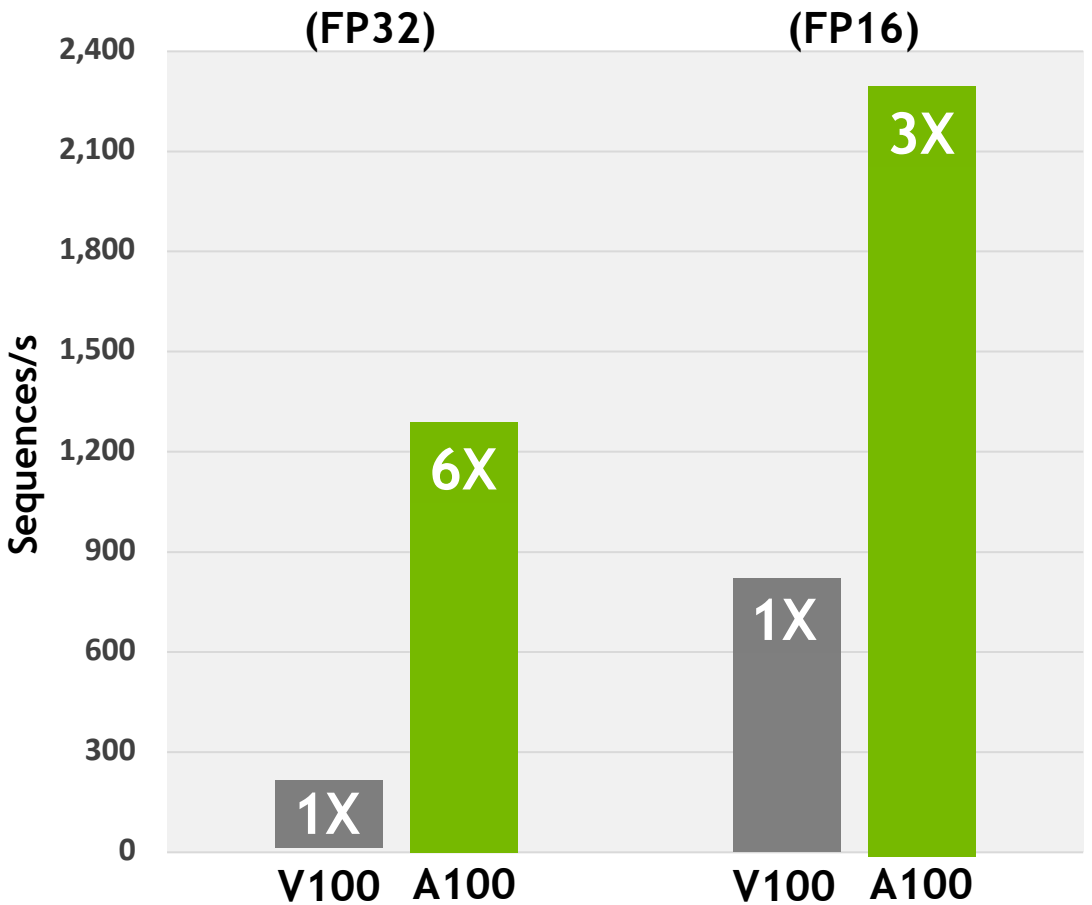
MIG



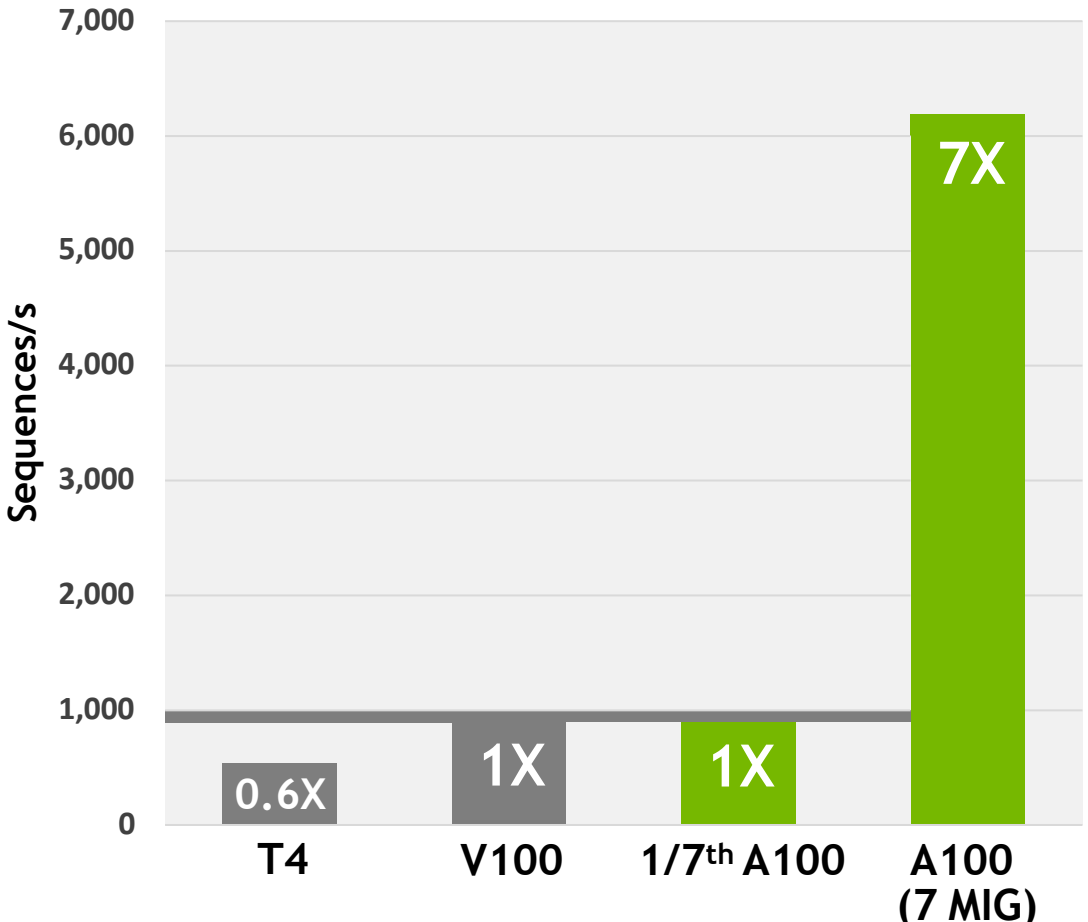
3rd GEN
NVLINK & NVSWITCH

UNIFIED AI ACCELERATION

BERT-LARGE TRAINING

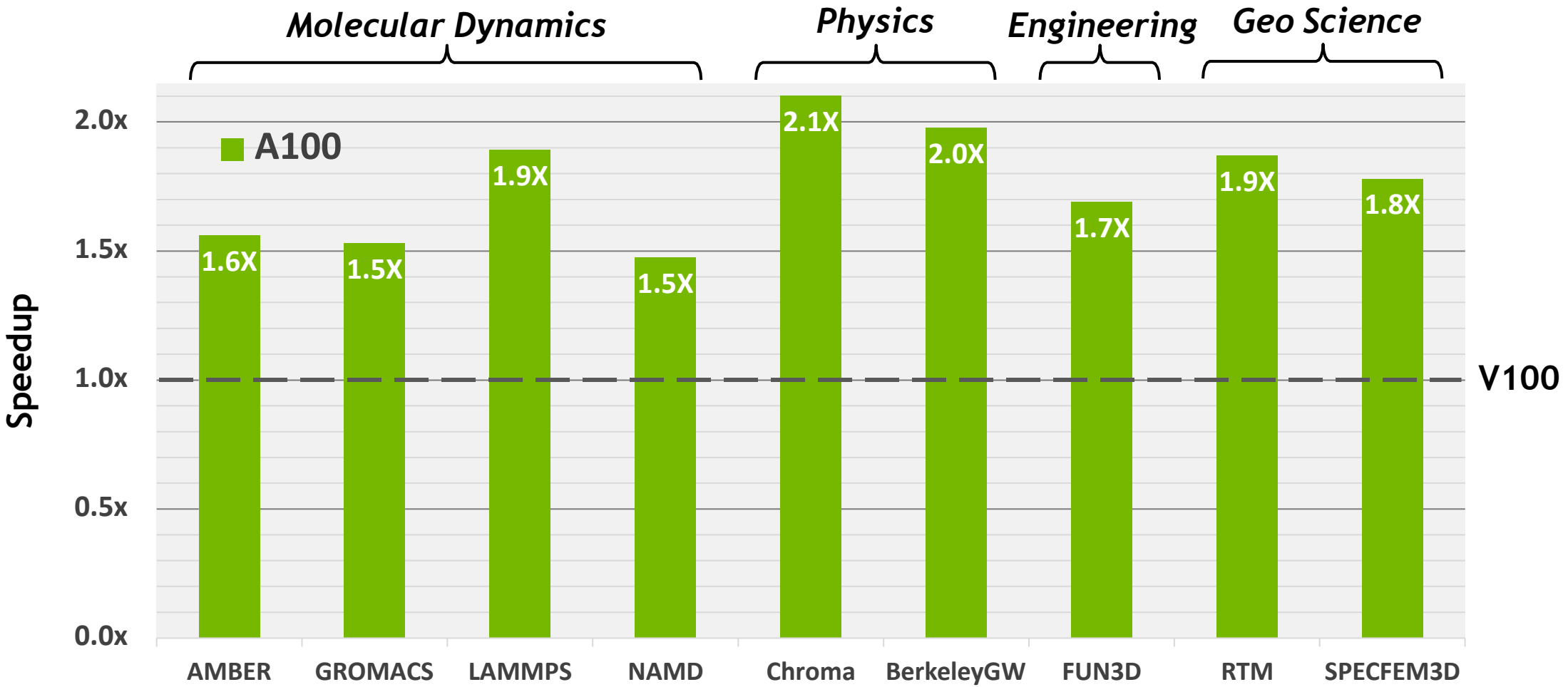


BERT-LARGE INFERENCE



All results are measured
BERT Large Training (FP32 & FP16) measures Pre-Training phase, uses PyTorch including (2/3) Phase1 with Seq Len 128 and (1/3) Phase 2 with Seq Len 512, V100 is DGX1 Server with 8xV100, A100 is DGX A100 Server with 8xA100, A100 uses TF32 Tensor Core for FP32 training
BERT Large Inference uses TRT 7.1 for T4/V100, with INT8/FP16 at batch size 256. Pre-production TRT for A100, uses batch size 94 and INT8 with sparsity

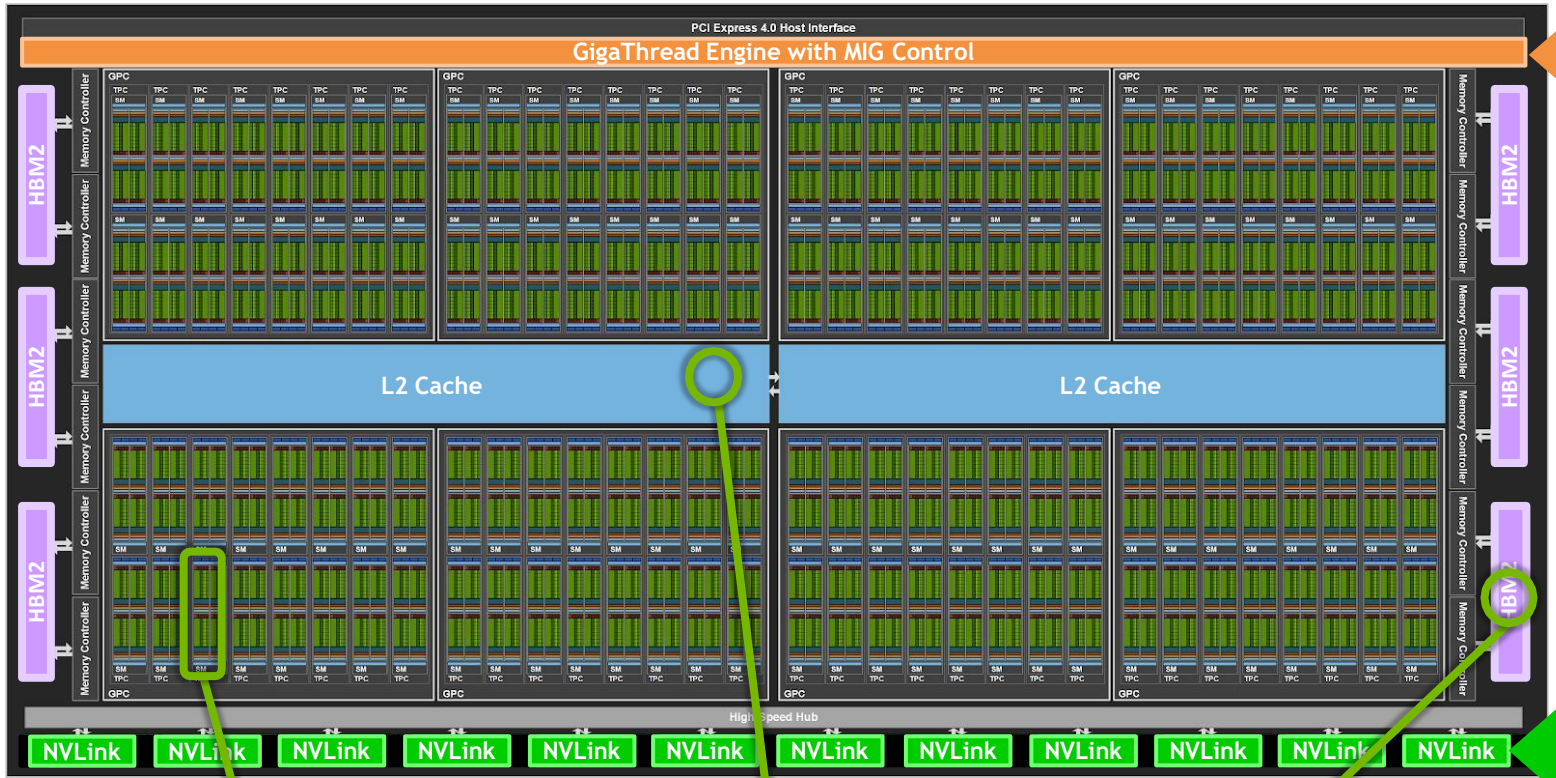
ACCELERATING HPC



All results are measured
Except BerkeleyGW, V100 used is single V100 SXM2. A100 used is single A100 SXM4
More apps detail: AMBER based on PME-Cellulose, GROMACS with STMV (h-bond), LAMMPS with Atomic Fluid LJ-2.5, NAMD with v3.0a1 STMV_NVE
Chroma with szscl21_24_128, FUN3D with dpw, RTM with Isotropic Radius 4 1024^3, SPECFEM3D with Cartesian four material model
BerkeleyGW based on Chi Sum and uses 8xV100 in DGX-1, vs 8xA100 in DGX A100

A100 TENSOR-CORE GPU

54 billion transistors in 7nm



7x

Scale OUT

Multi-Instance GPU

2x BW

Scale UP

3rd gen. NVLINK

108 SMs
6912 CUDA Cores

40MB L2
6.7x capacity

1.56 TB/s HBM2
1.7x bandwidth

A100 SM

Third-generation Tensor Core

Faster and more efficient
Comprehensive data types
Sparsity acceleration

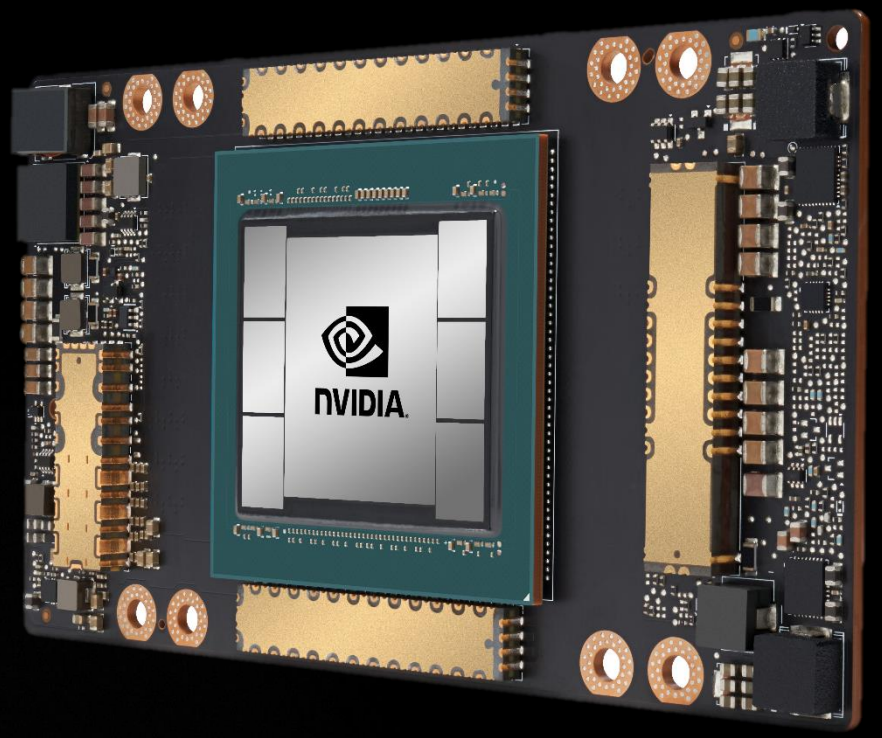
Asynchronous data movement and synchronization

Increased L1/SMEM capacity

INTRODUCING NVIDIA A100

Greatest Generational Leap – 20X Volta

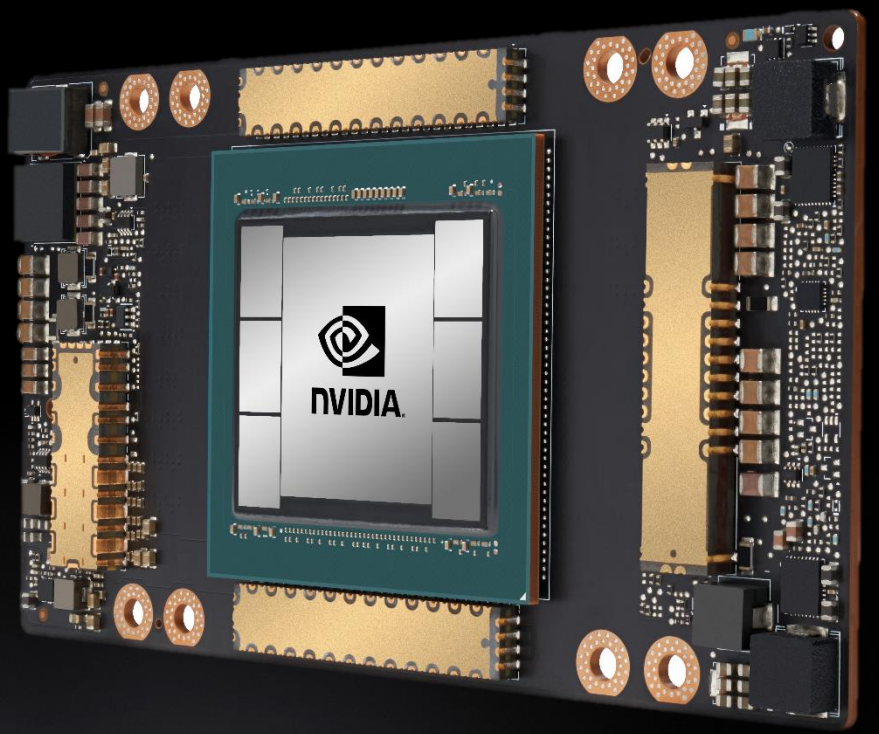
	Peak	Vs Volta
FP32 TRAINING	312 TFLOPS	20X
INT8 INFERENCE	1,248 TOPS	20X
FP64 HPC	19.5 TFLOPS	2.5X
MULTI INSTANCE GPU		7X GPUs



54B XTOR | 826mm2 | TSMC 7N | 40GB Samsung HBM2 | 600 GB/s NVLink

NVIDIA A100 DETAILED SPECS

	Peak Performance
Transistor Count	54 billion
Die Size	826 mm^2
FP64 CUDA Cores	3,456
FP32 CUDA Cores	6,912
Tensor Cores	432
Streaming Multiprocessors	108
FP64	9.7 teraFLOPS
FP64 Tensor Core	19.5 teraFLOPS
FP32	19.5 teraFLOPS
TF32 Tensor Core	156 teraFLOPS | 312 teraFLOPS*
BFLOAT16 Tensor Core	312 teraFLOPS | 624 teraFLOPS*
FP16 Tensor Core	312 teraFLOPS | 624 teraFLOPS*
INT8 Tensor Core	624 TOPS | 1,248 TOPS*
INT4 Tensor Core	1,248 TOPS | 2,496 TOPS*
GPU Memory	40 GB
Interconnect	NVLink 600 GB/s PCIe Gen4 64 GB/s
Multi-Instance GPUs	Various Instance sizes with up to 7MIGs @5GB
Form Factor	4/8/16 SXM GPUs in HGX A100
Max Power	400W (SXM)

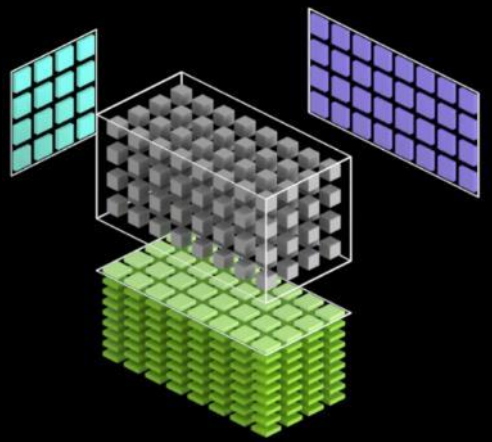


* Includes Sparsity

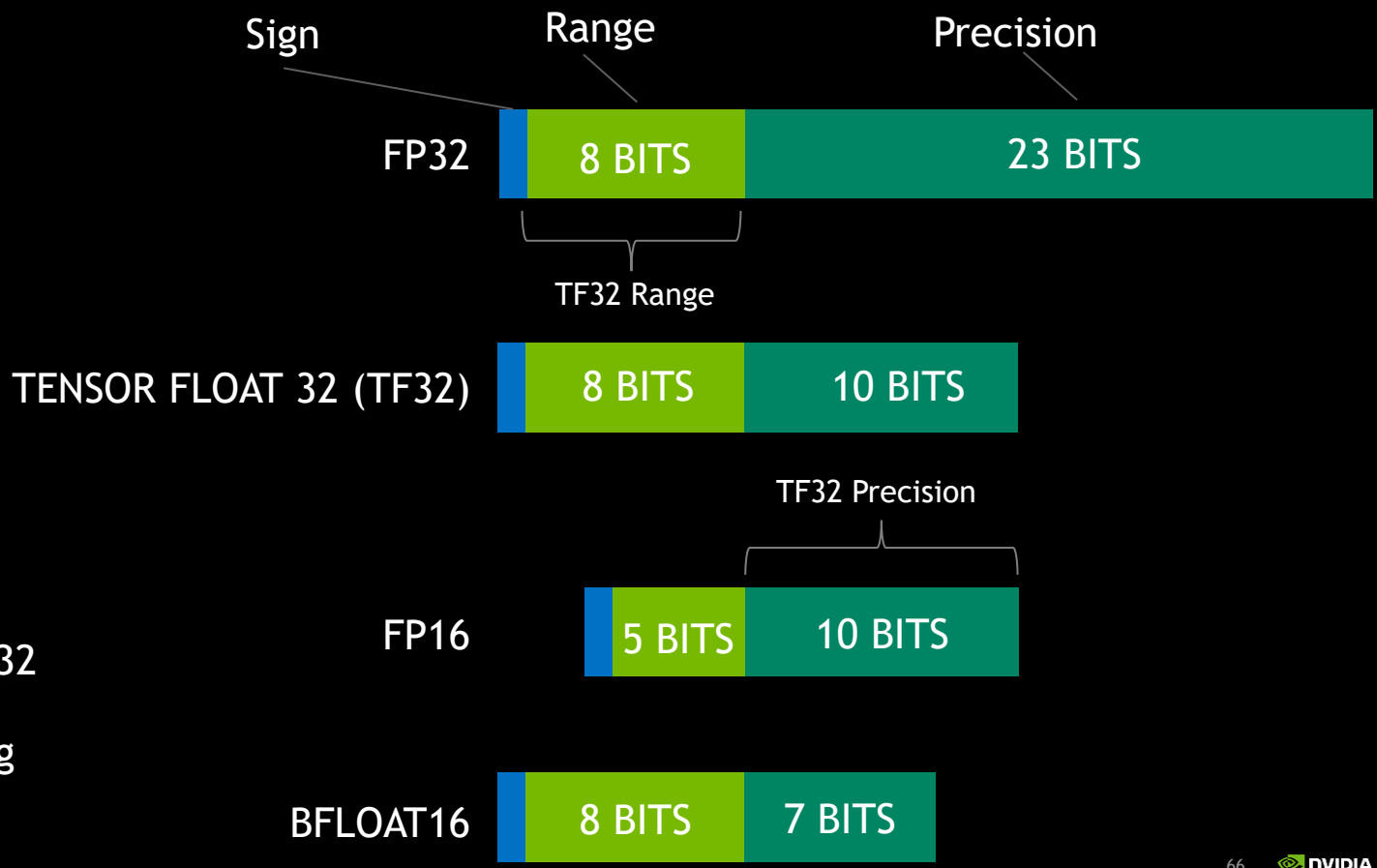

1. New Tensor Core
2. Strong Scaling
3. Elastic GPU
4. Productivity

1. New Tensor Core
2. Strong Scaling
3. Elastic GPU
4. Productivity

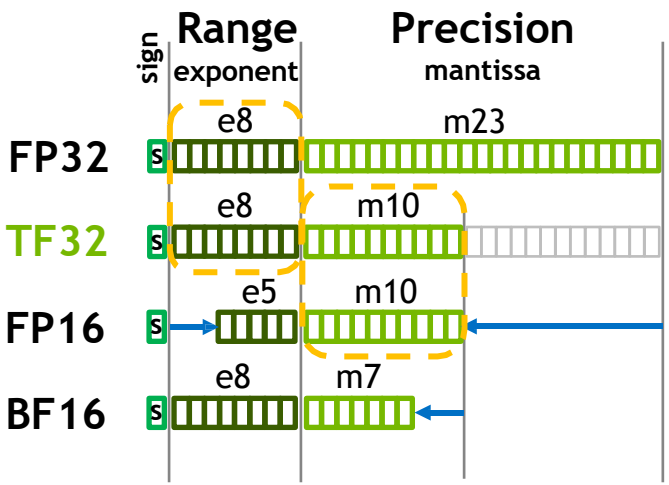
NEW TF32 TENSOR CORES



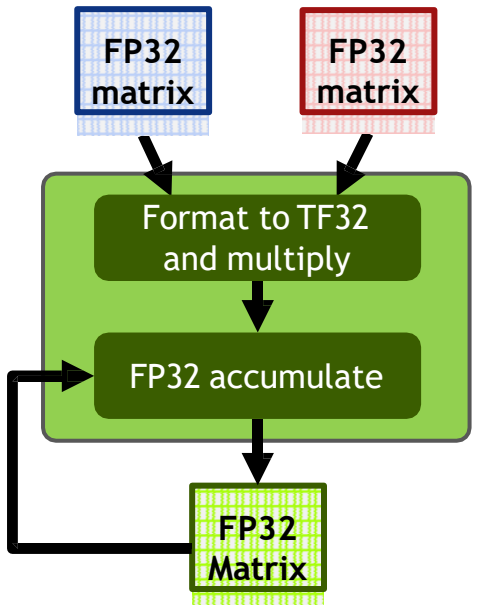
- Range of FP32 and Precision of FP16
- Input in FP32 and Accumulation in FP32
- No Code Change Speed-up for Training



INSIDE A100 TensorFloat-32 (TF32)



Range of FP32 with precision of FP16



FP32 input/output

FP32 storage and math for all activations, gradients, …
everything outside tensor cores

Out-of-the-box tensor core acceleration for DL

Easy step towards maximizing tensor core performance with mixed-precision (FP16, BF16)

Up to 4x speedup on linear solvers for HPC

S22082: Mixed-Precision Training of Neural Networks, 5/20 2:45pm PDT

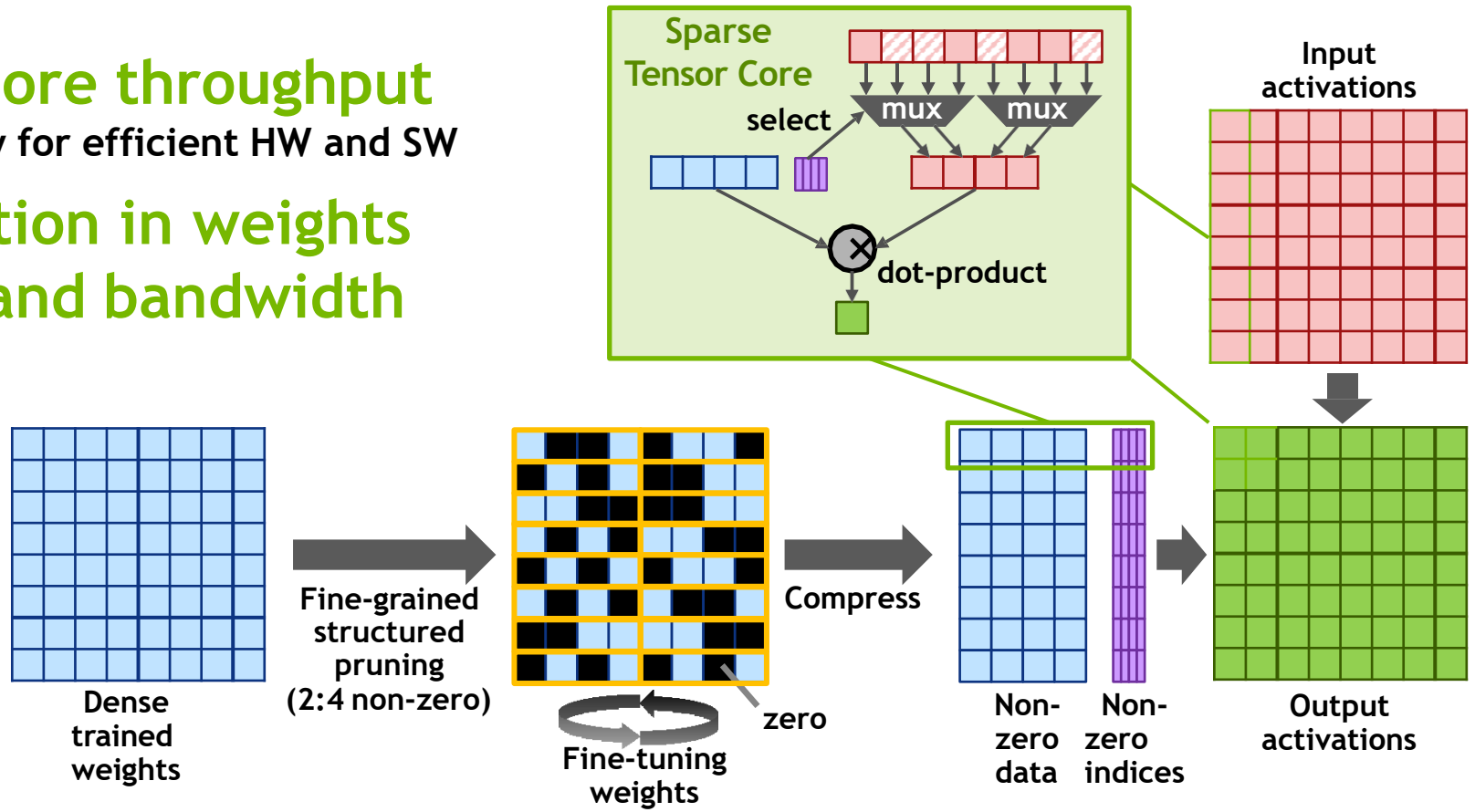
S21681: How CUDA Math Libraries can help you unleash the power of the new NVIDIA A100 GPU (recording available)

INSIDE A100 SPARSE TENSOR CORE

2x Tensor Core throughput

Structured-sparsity for efficient HW and SW

~2x reduction in weights footprint and bandwidth



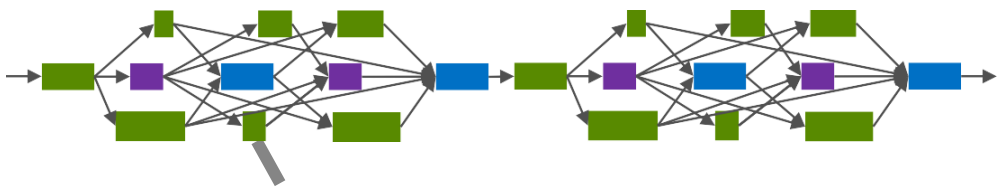
~No loss in inferencing accuracy

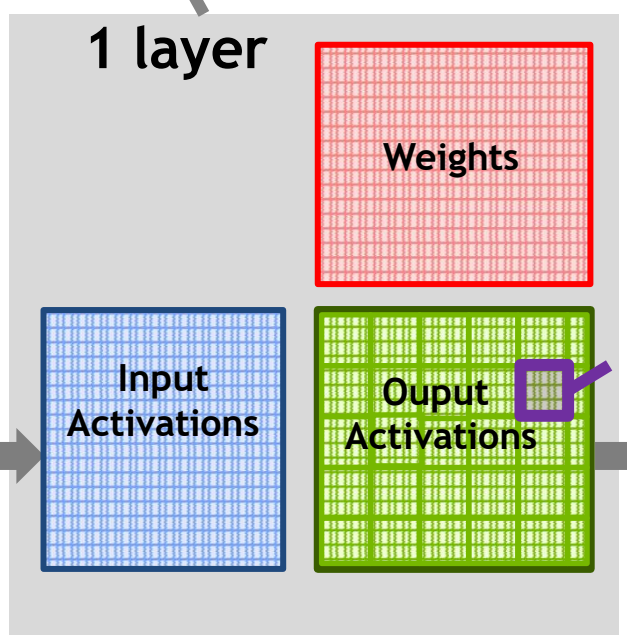
Evaluated across dozens of networks: vision, object detection, segmentation, natural language modeling, translation

1. New Tensor Core
2. Strong Scaling
3. Elastic GPU
4. Productivity

DL STRONG SCALING

DL networks:
Long chains of sequentially-dependent compute-intensive layers

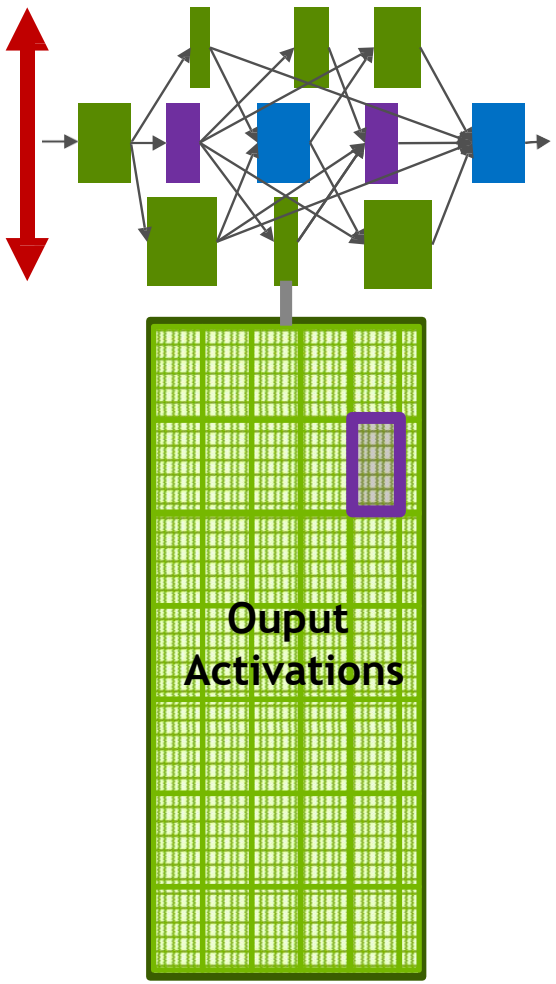




Each layer is parallelized across GPU

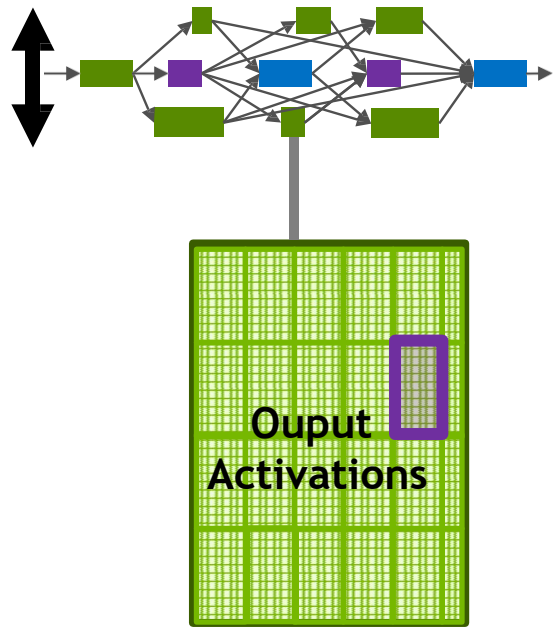
Tile: work for 1 SM

Weak scaling



~2.5x larger network runs in same time

Strong scaling

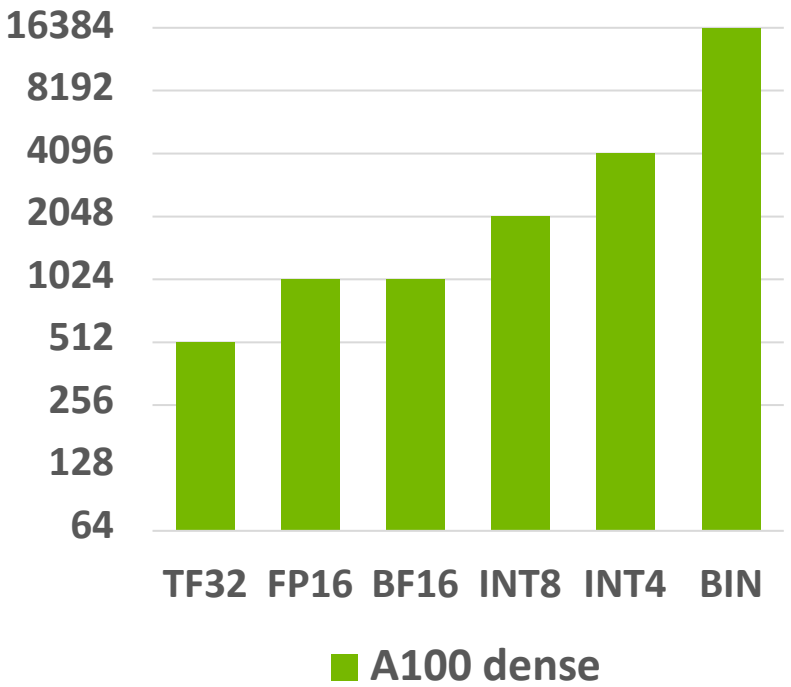


Fixed network runs ~2.5x faster

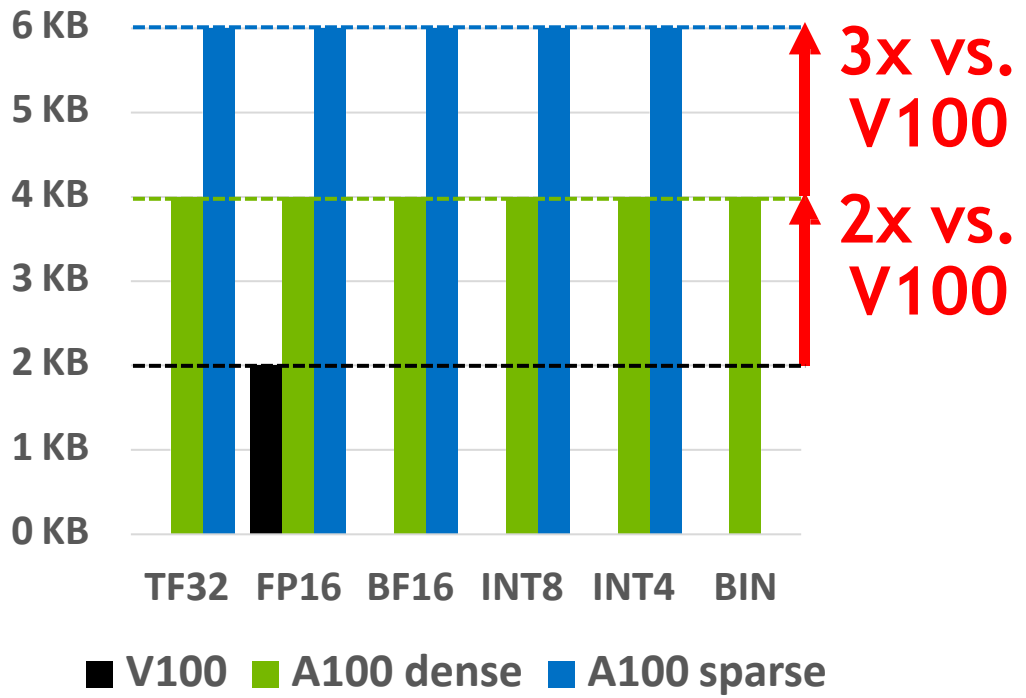
NVIDIA

HOW TO KEEP TENSOR CORES FED?

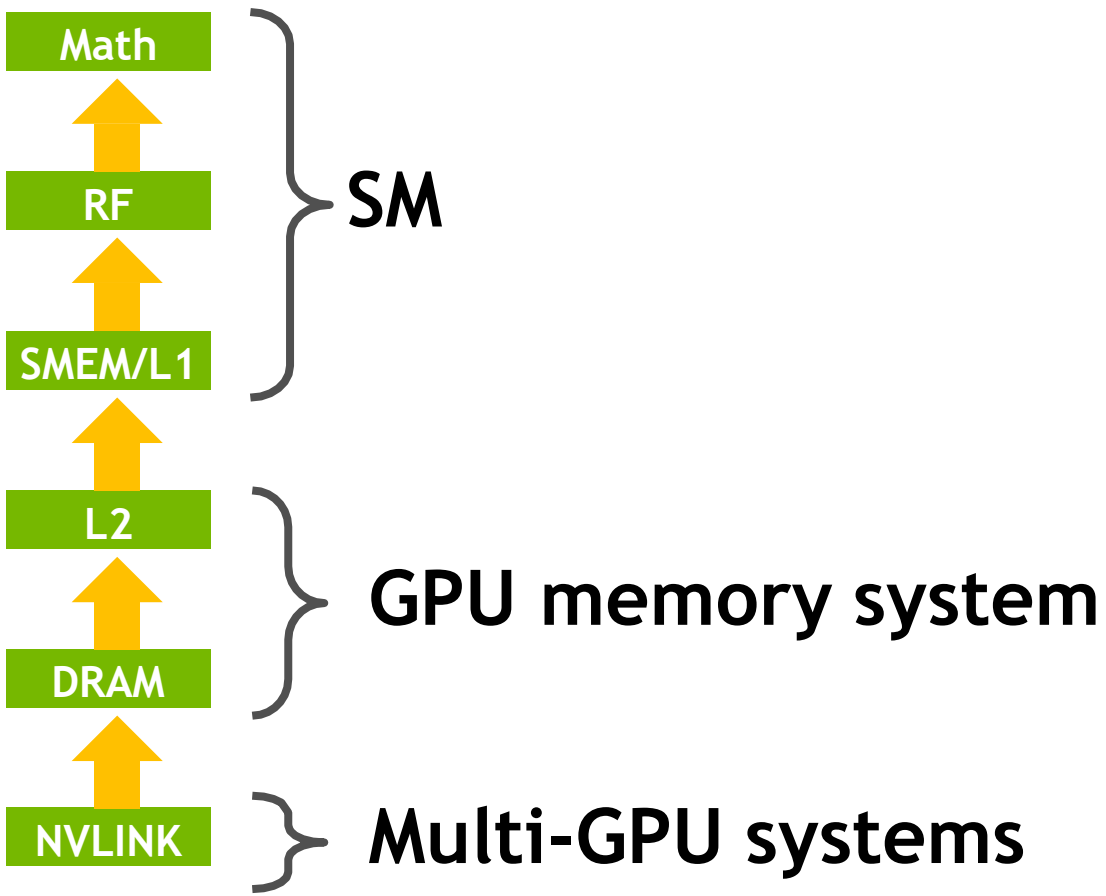
Math bandwidth
(MACs/clock/SM)



Required data bandwidth
(A+B operands, B/clock/SM)



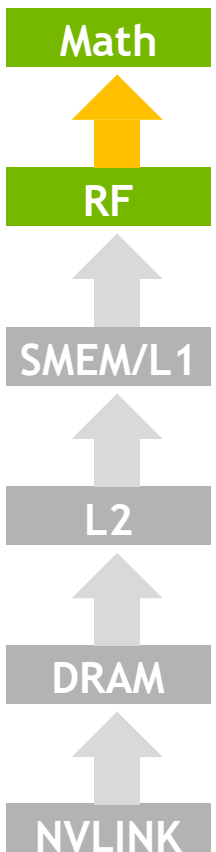
A100 STRONG SCALING INNOVATIONS

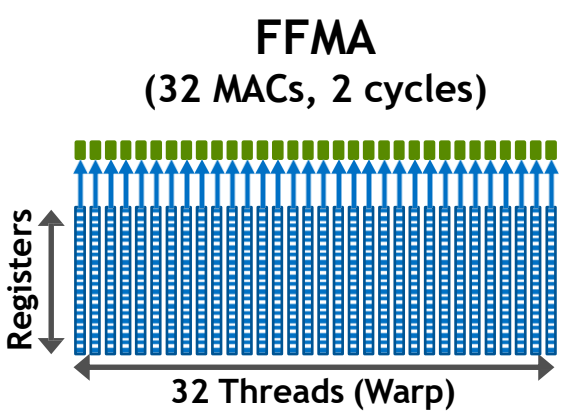


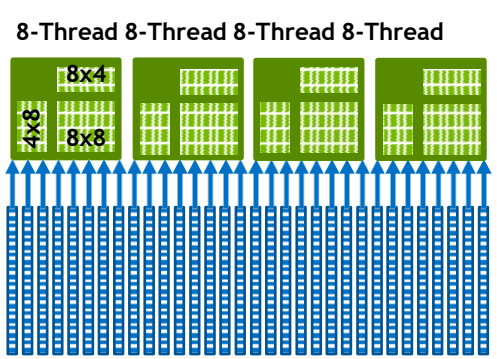
Improve speeds & feeds and efficiency across all levels of compute and memory hierarchy

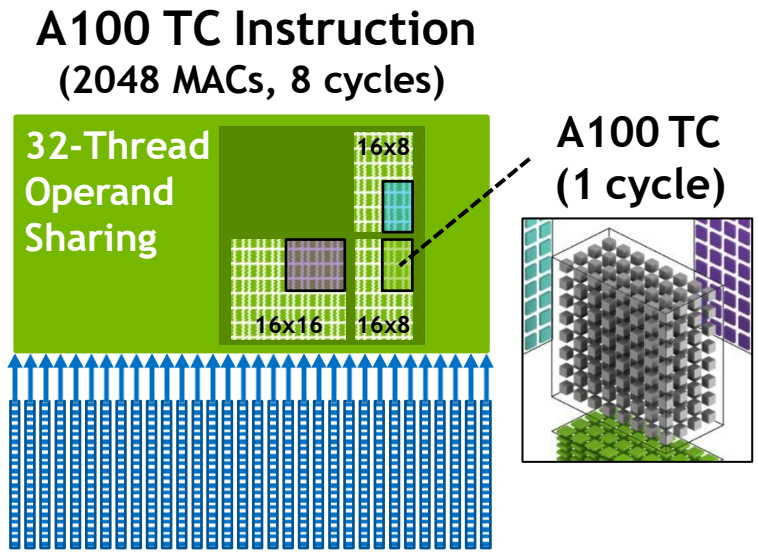
A100 TENSOR CORE

2x throughput vs. V100, >2x efficiency







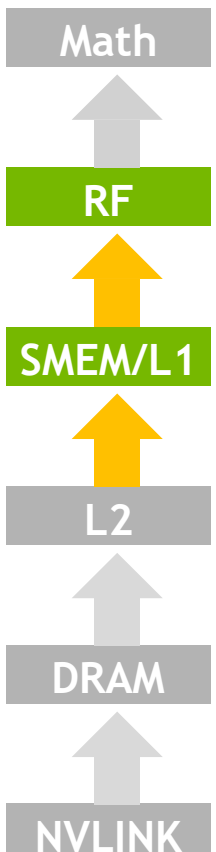


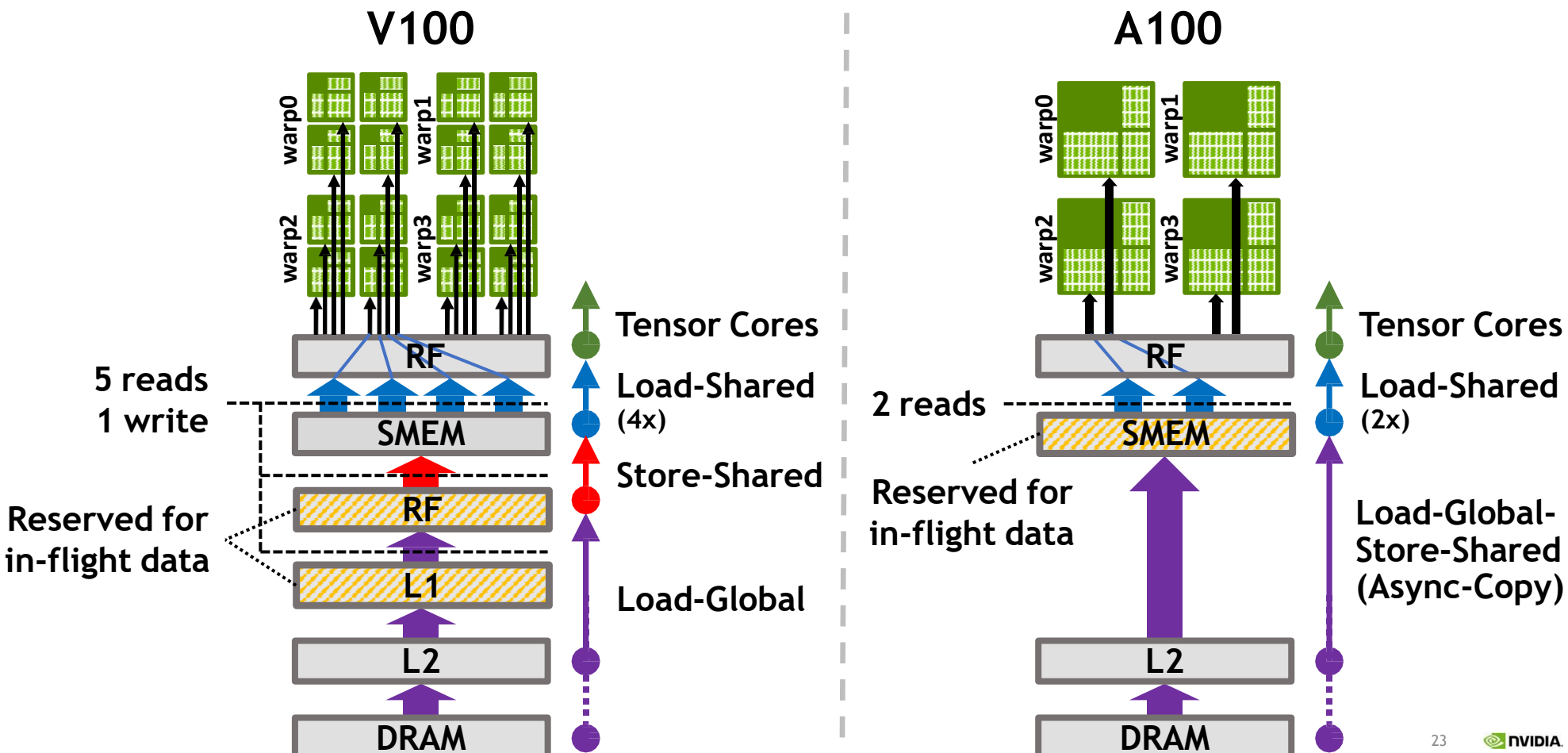
16x16x16 matrix multiply	FFMA	V100 TC	A100 TC	A100 vs. V100 (improvement)	A100 vs. FFMA (improvement)
Thread sharing	1	8	32	4x	32x
Hardware instructions	128	16	2	8x	64x
Register reads+writes (warp)	512	80	28	2.9x	18x
Cycles	256	32	16	2x	16x

Tensor Cores assume FP16 inputs with FP32 accumulator, V100 Tensor Core instruction uses 4 hardware instructions

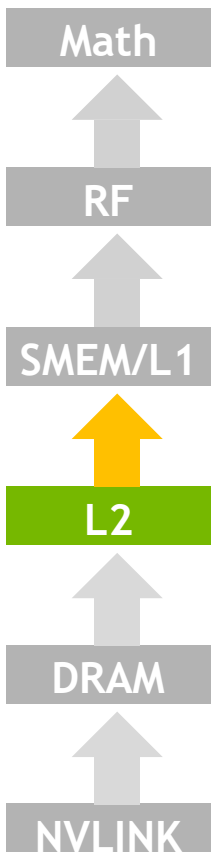
A100 SM DATA MOVEMENT EFFICIENCY

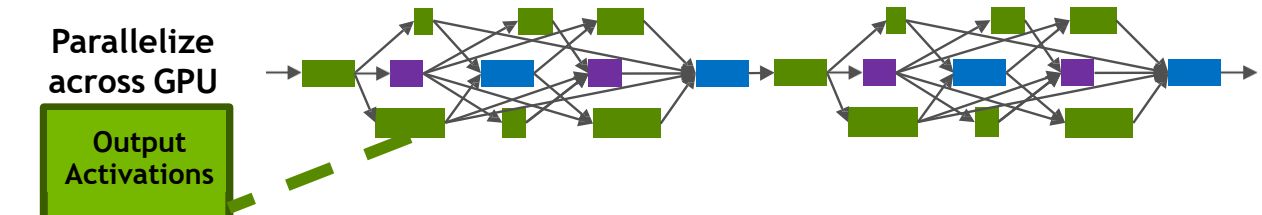
3x SMEM/L1 bandwidth, 2x in-flight capacity

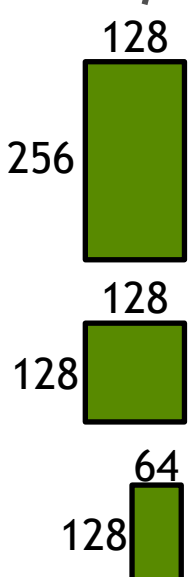




A100 L2 BANDWIDTH

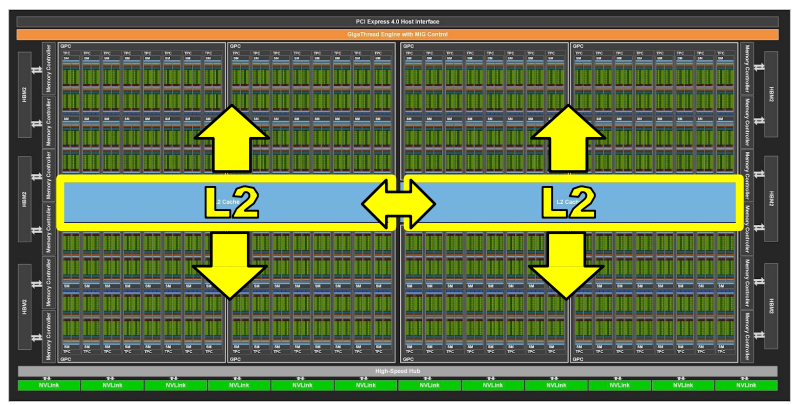






	V100	V100++ (hypothetical)	A100
	80 SMs V100 TC 64 L2 slices 32 B/clk/slice	108 SMs A100 TC 64 L2 slices 32 B/clk/slice	108 SMs A100 TC 80 L2 slices 64 B/clk/slice
	12 B/clk/SM 47%	24 B/clk/SM 127%	24 B/clk/SM 51%
	16 B/clk/SM 63%	32 B/clk/SM 169%	32 B/clk/SM 68%
	24 B/clk/SM 94%	48 B/clk/SM 253%	48 B/clk/SM 101%

Split L2 with hierarchical crossbar - 2.3x increase in bandwidth over V100, lower latency



A100 DRAM BANDWIDTH

Faster HBM2

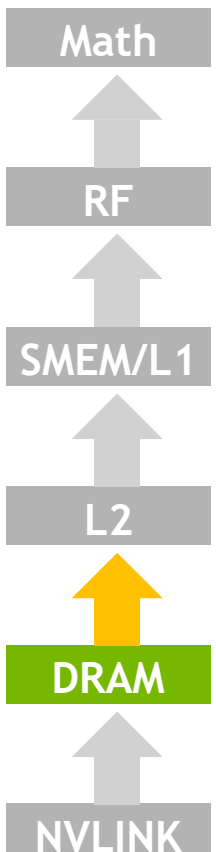
25% more pins, 38% faster clocks

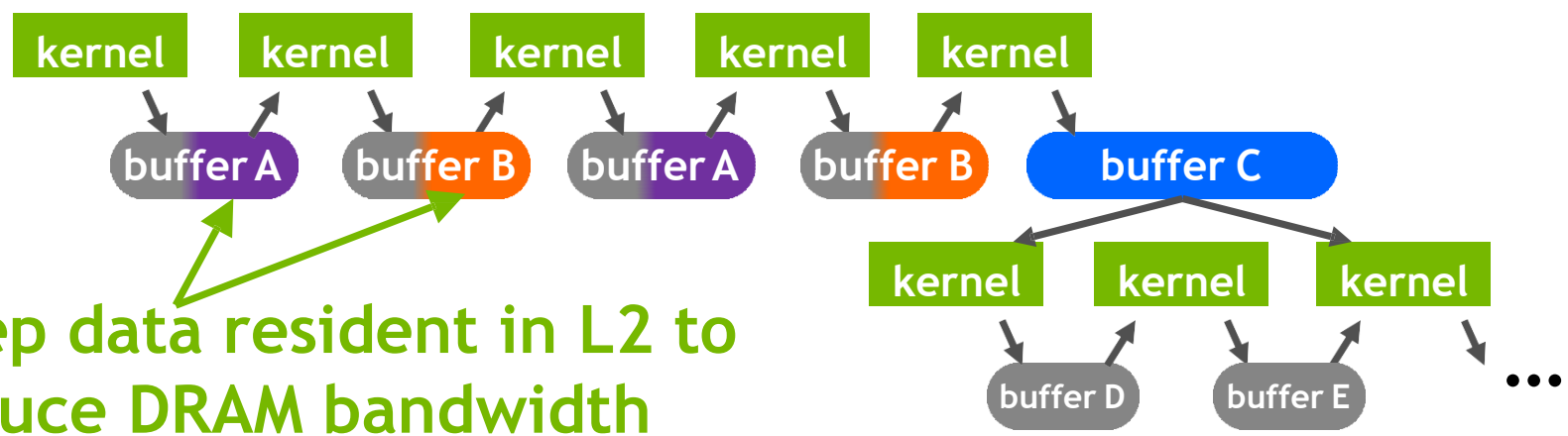
❼ 1.6 TB/s, 1.7x vs. V100

Larger and smarter L2

40MB L2, 6.7x vs. V100

L2-Residency controls

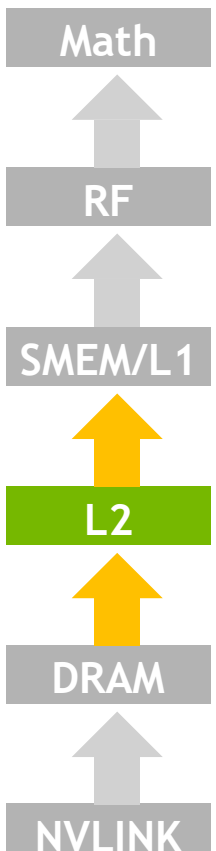




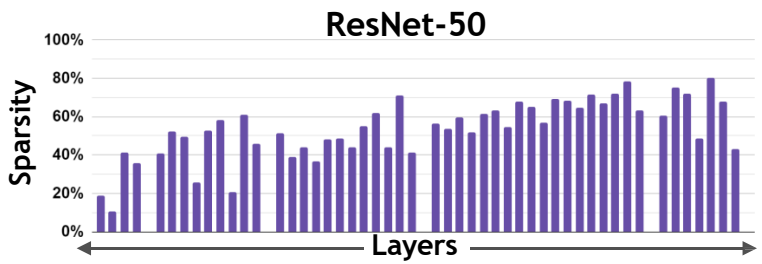
Keep data resident in L2 to reduce DRAM bandwidth

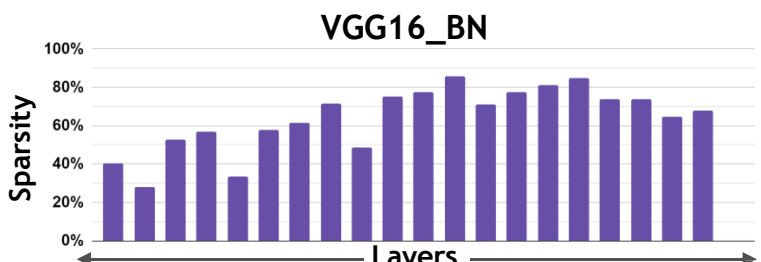
❼ S21819: Optimizing Applications for NVIDIA Ampere GPU Architecture, 5/21 10:15am PDT

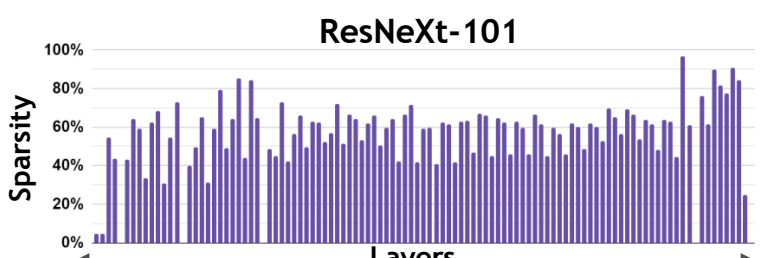
A100 COMPUTE DATA COMPRESSION



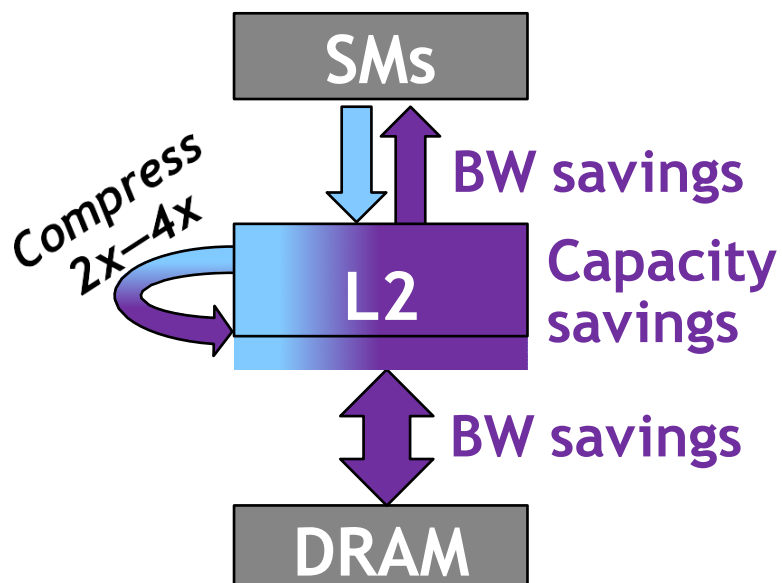
Activation sparsity due to ReLU







Up to 4x DRAM+L2 bandwidth and 2x L2 capacity for fine-grained unstructured sparsity



❼ S21819: Optimizing Applications for NVIDIA Ampere GPU Architecture, 5/21 10:15am PDT

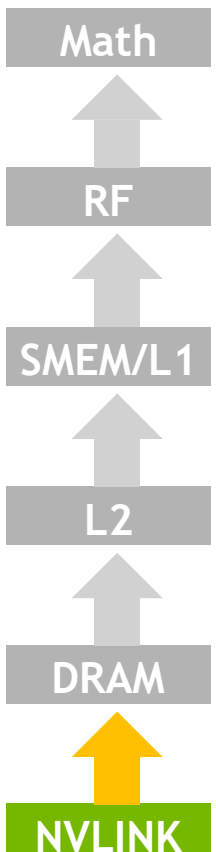
A100 NVLINK BANDWIDTH

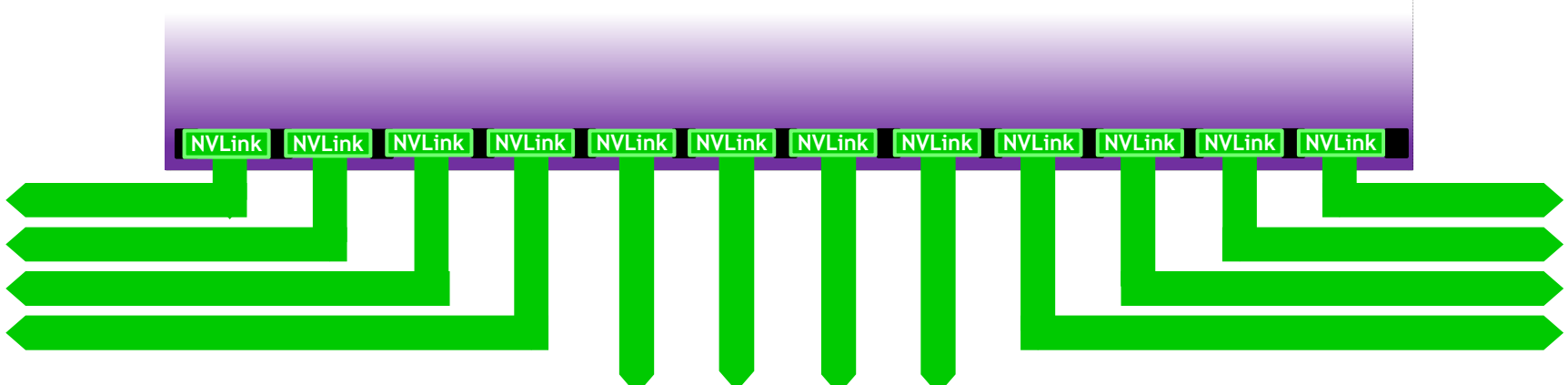
Third Generation NVLink

50 Gbit/sec per signal pair

12 links, 25 GB/s in/out, 600 GB/s total

2x vs. V100

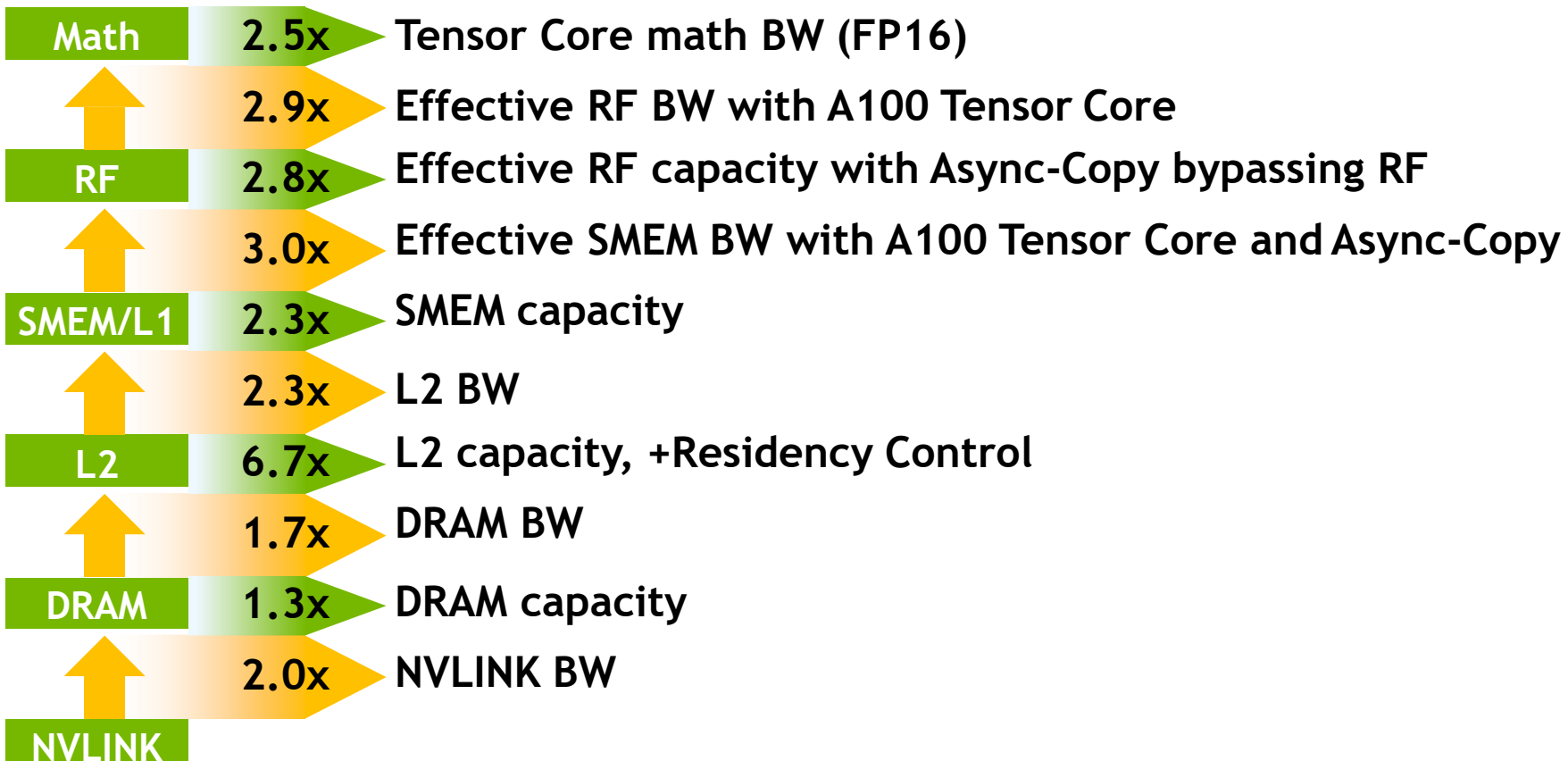




A100 STRONG SCALING INNOVATIONS

Delivering unprecedented levels of performance

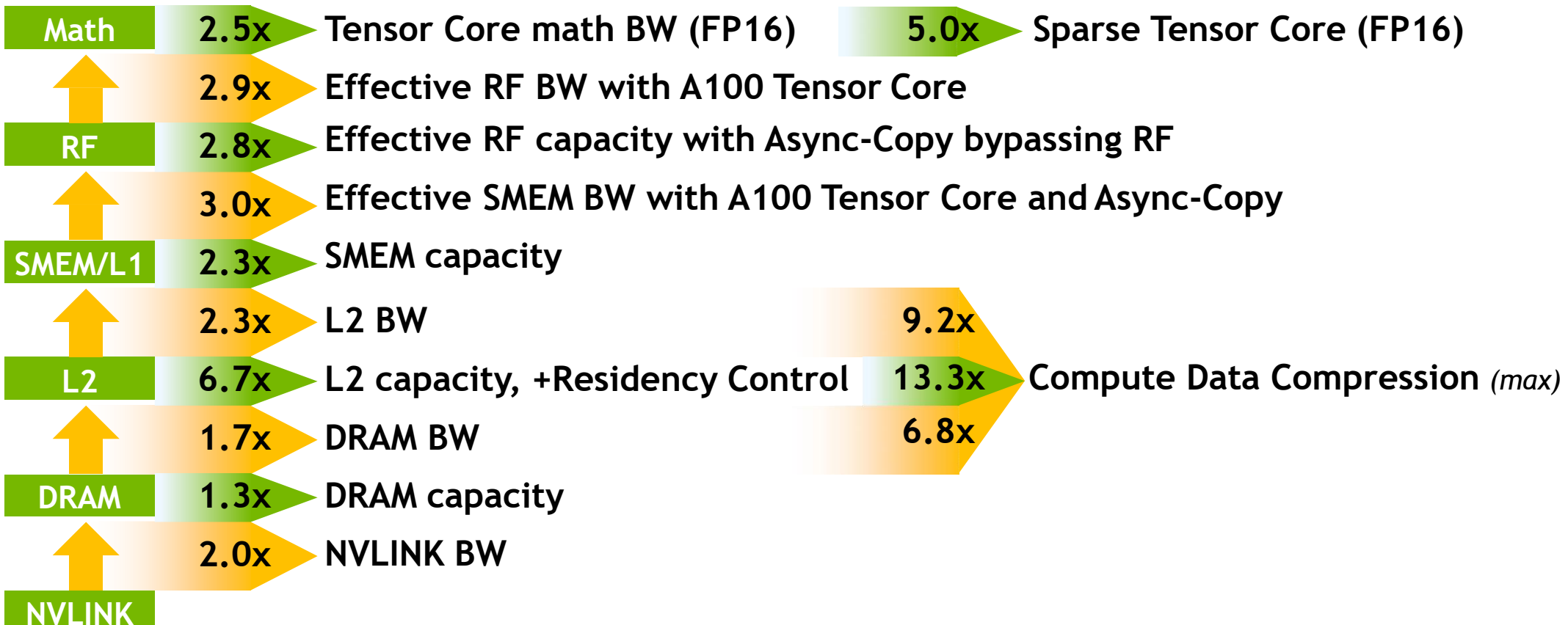
A100 improvements over V100



A100 STRONG SCALING INNOVATIONS

Delivering unprecedented levels of performance

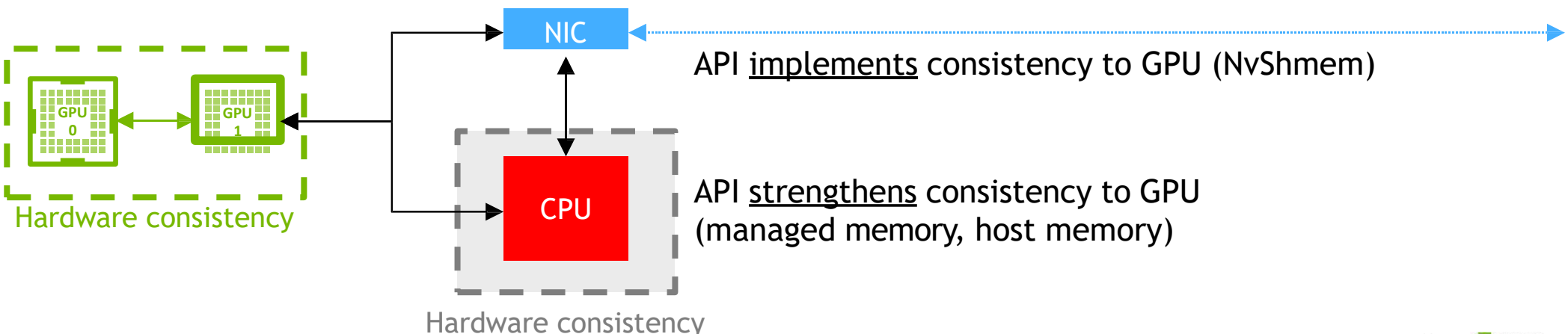
A100 improvements over V100



1. New Tensor Core
2. Strong Scaling
3. **Elastic GPU**
4. Productivity

NVLINK: ONE BIG GPU

- **InfiniBand/Ethernet**: travels a long distance, consistency is the responsibility of software
- **PCI Express**: hardware consistency for I/O, not for programming language memory models
- **NVLINK**: hardware consistency for programming language memory models, like system bus

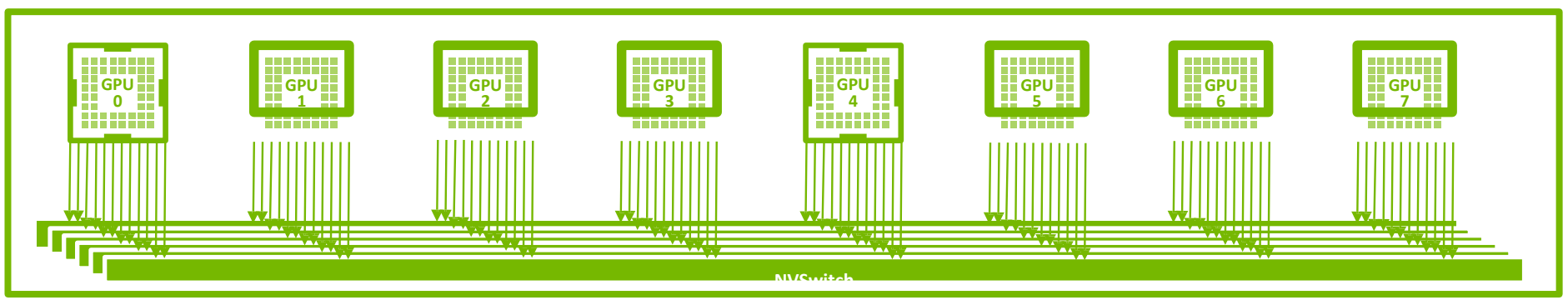


HGX A100: 3RD GEN NVLINK

- **HGX A100 4-GPU:** fully-connected system with 100GB/s all-to-all BW

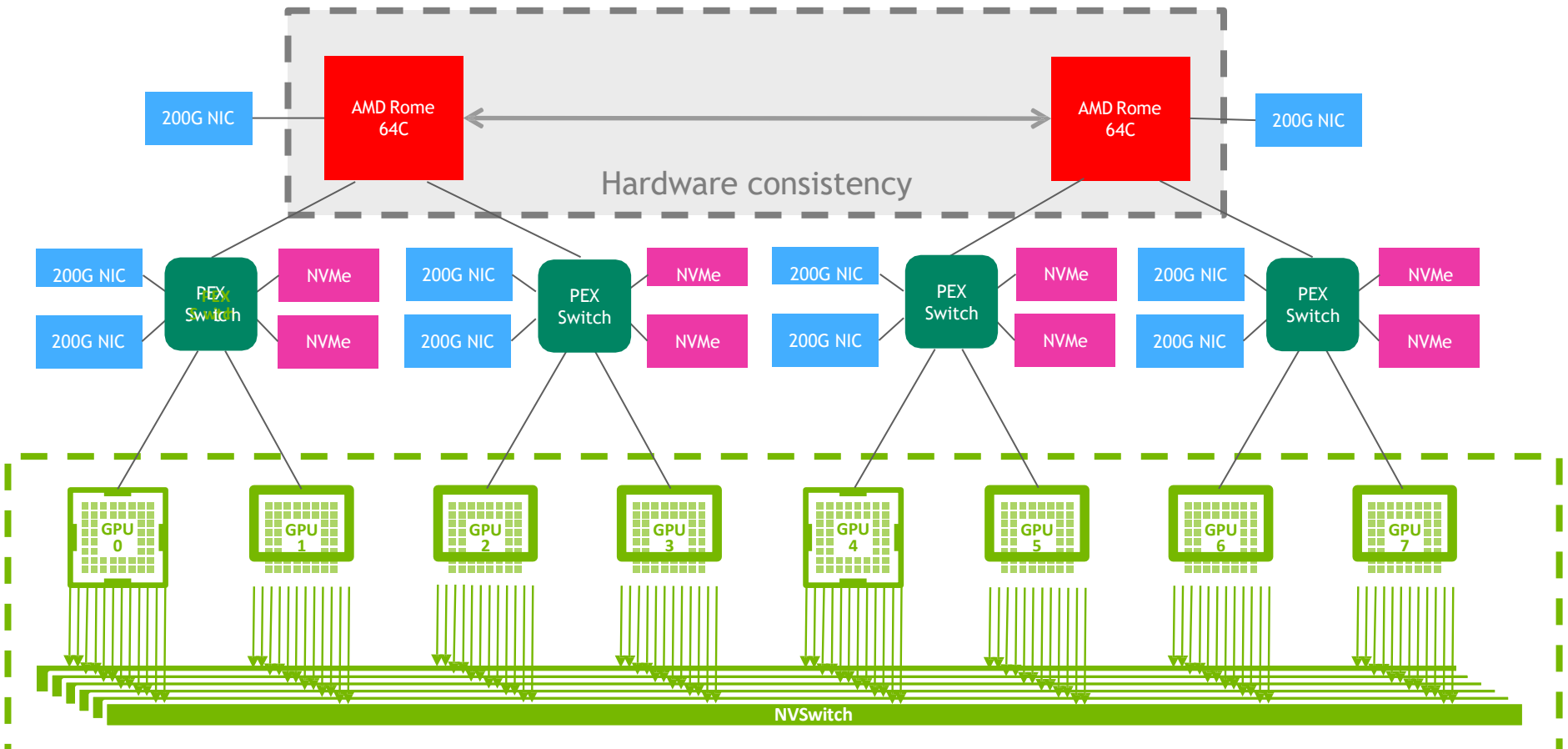
HGX A100: 3RD GEN NVLINK & SWITCH

- **HGX A100 4-GPU:** fully-connected system with 100GB/s all-to-all BW
- **New NVSwitch:** 6B transistors in TSMC 7FF, 36 ports, 25GB/s each, per direction
- **HGX A100 8-GPU:** 6x NVSwitch in a fat tree topology, 2.4TB/s full-duplex bandwidth



Hardware consistency

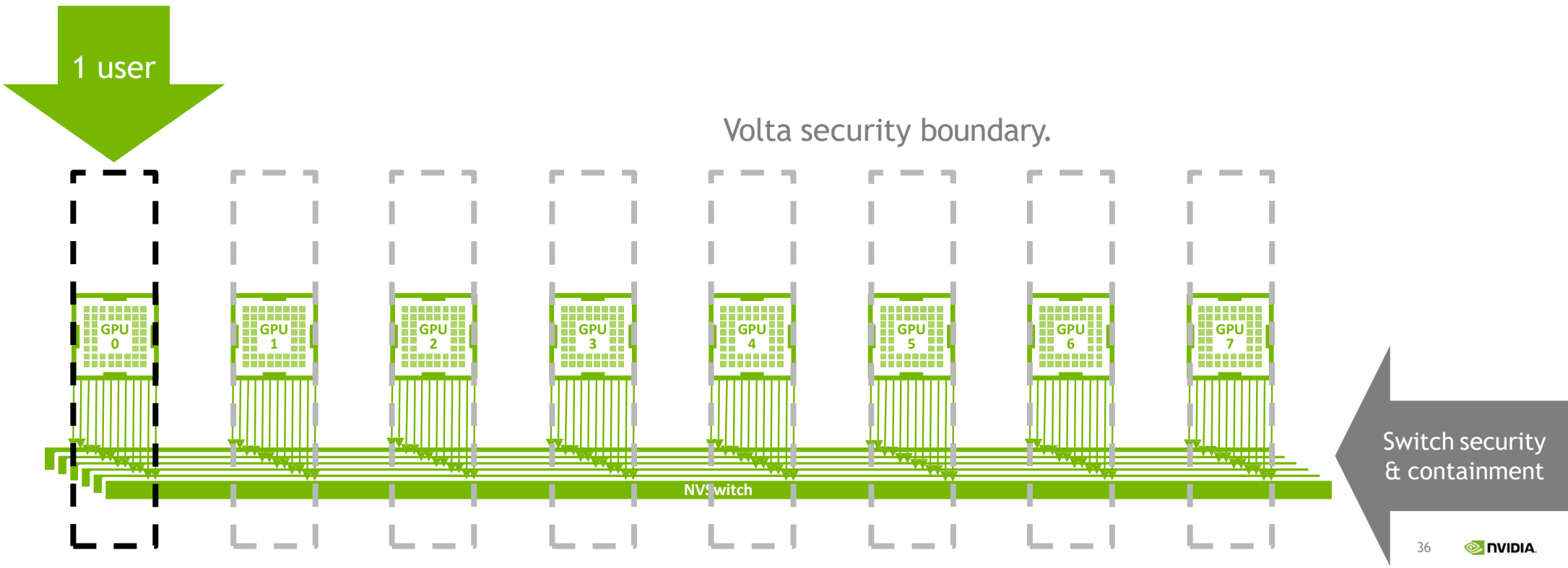
DGX A100: PCIE4 CONTROL & I/O



Hardware consistency

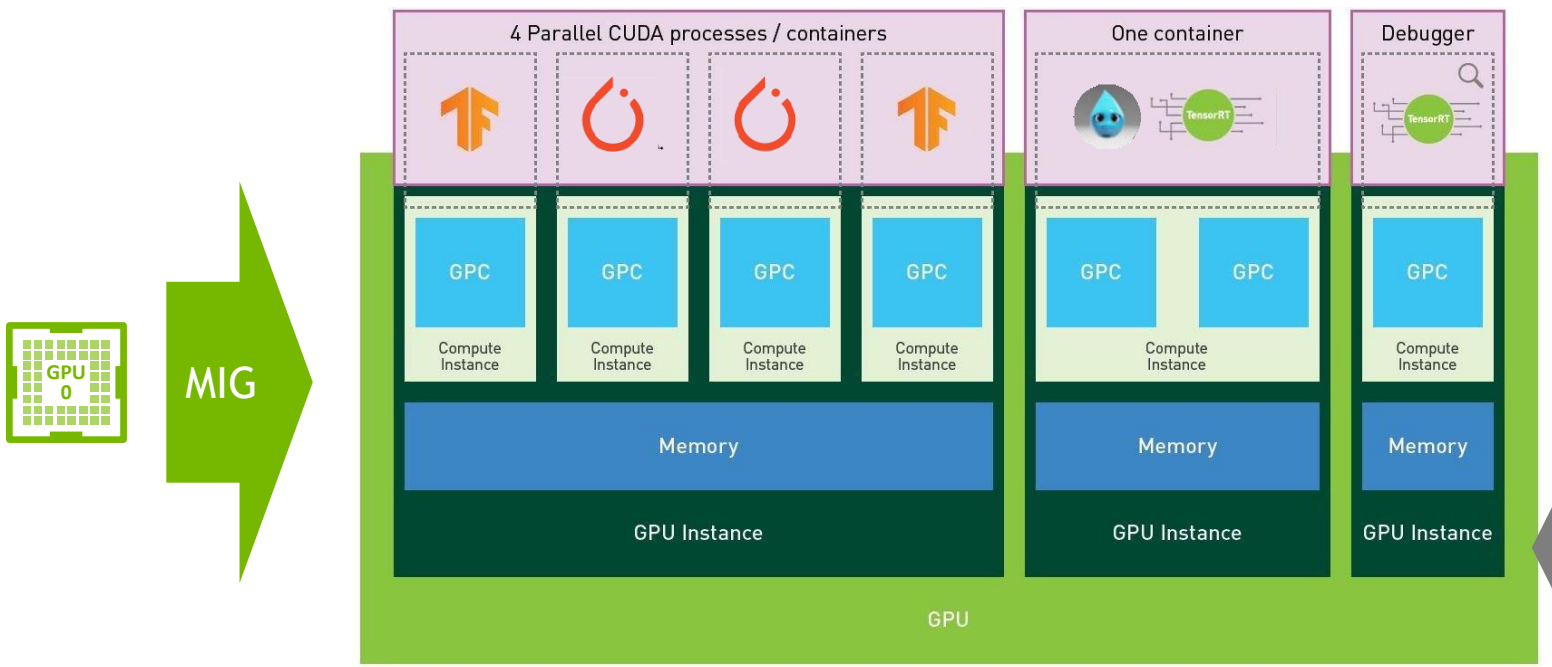
CLOUD SMALL INSTANCE USAGE

- Small workloads can under-utilize GPU cloud instances, provisioned at whole GPU level
- CSPs can't use MPS for GPU space-sharing, because it doesn't provide enough isolation



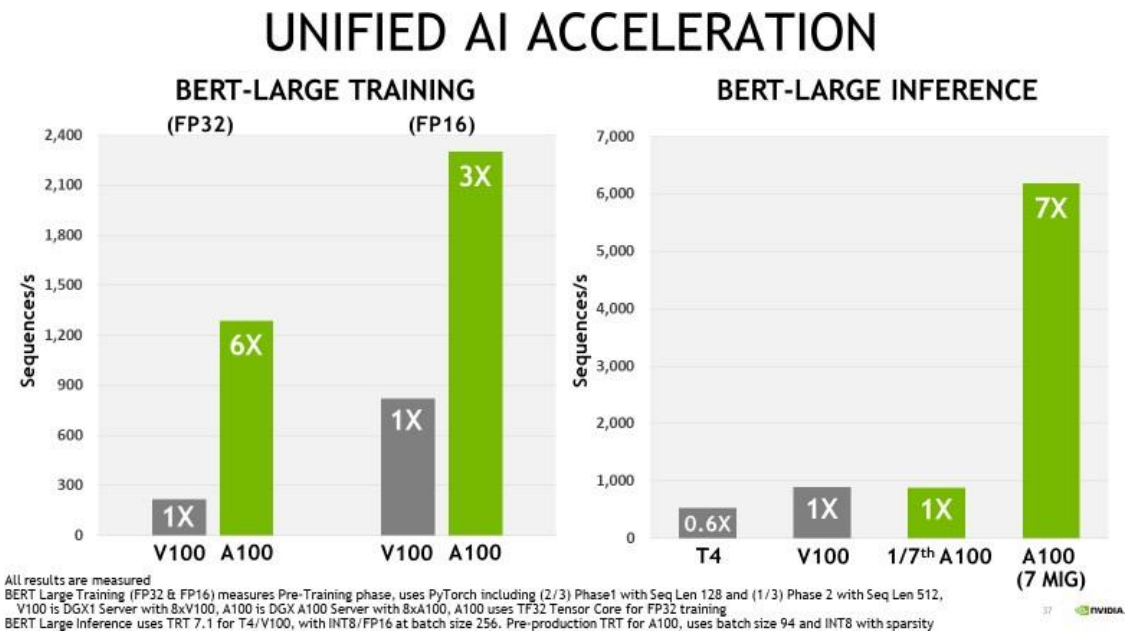
NEW: MULTI-INSTANCE GPU (MIG)

- Up to 7 instances total, dynamically reconfigurable
- **Compute instances:** compute/fault isolation, but share/compete for memory
- **GPU instances:** separate and isolated paths through the entire memory system



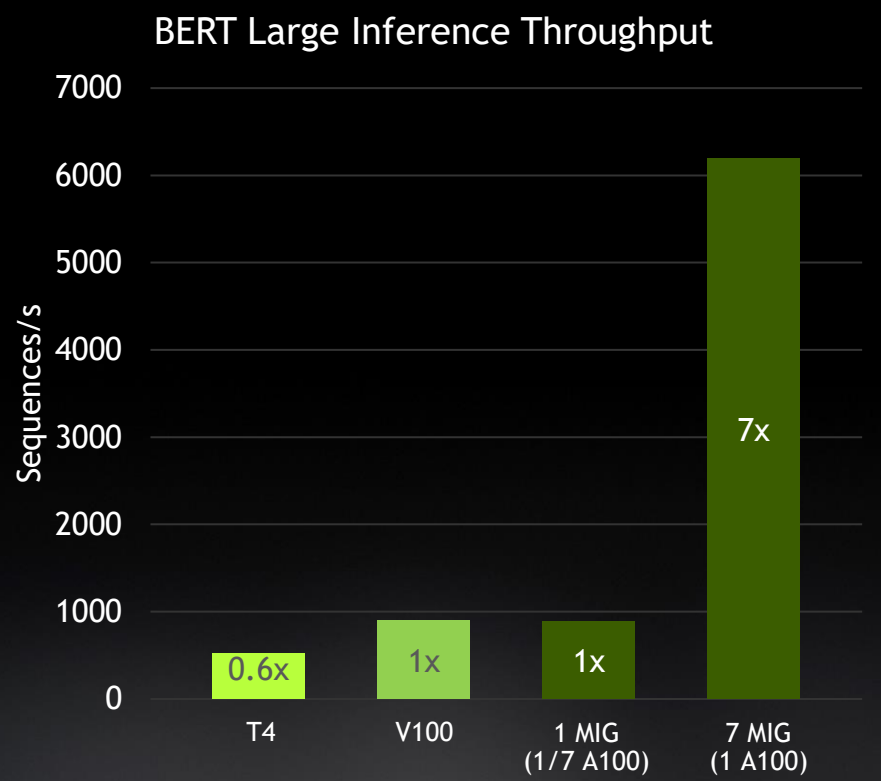
ELASTIC GPU COMPUTING

- Each A100 is 1 to 7 GPUs
- Each DGX A100 is 1 to 56 GPUs
- Each GPU can serve a different user, with full memory isolation and QoS



- S21975: Inside NVIDIA's Multi-Instance GPU Feature (recording available)
- S21884: Under the Hood of the new DGX A100 System Architecture (recording available soon)
- S21702: Introducing NVIDIA DGX A100: The Universal AI System for Enterprise, 5/20 9:00am PDT

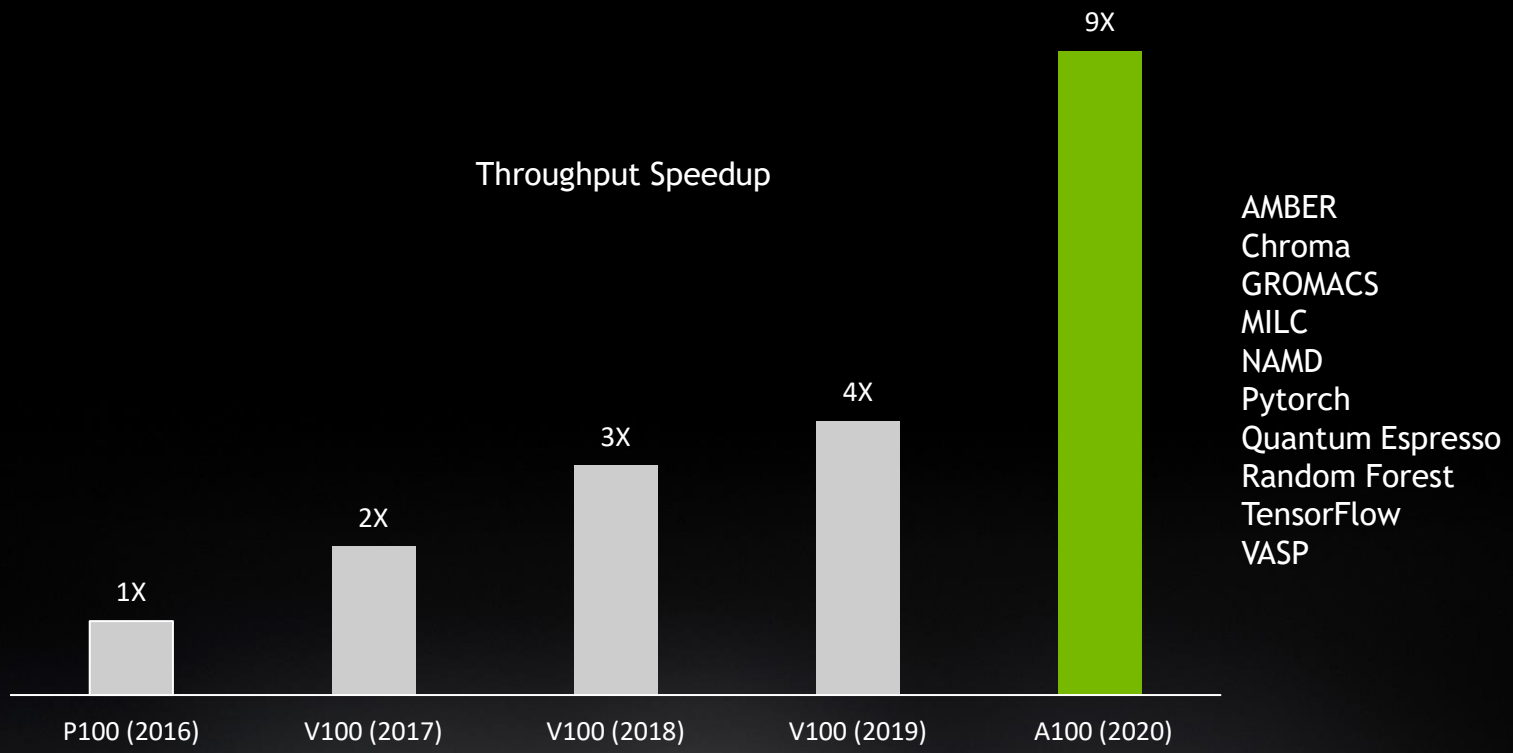
7X HIGHER INFERENCE THROUGHPUT WITH MIG



BERT Large Inference | T4: TRT 7.1, Precision = INT8, Batch Size =256, V100: TRT 7.1, Precision = FP16, Batch Size =256 | A100 with 7 MIG instances of 1g.5gb : Pre-production TRT, Batch Size =94, Precision = INT8 with Sparsity

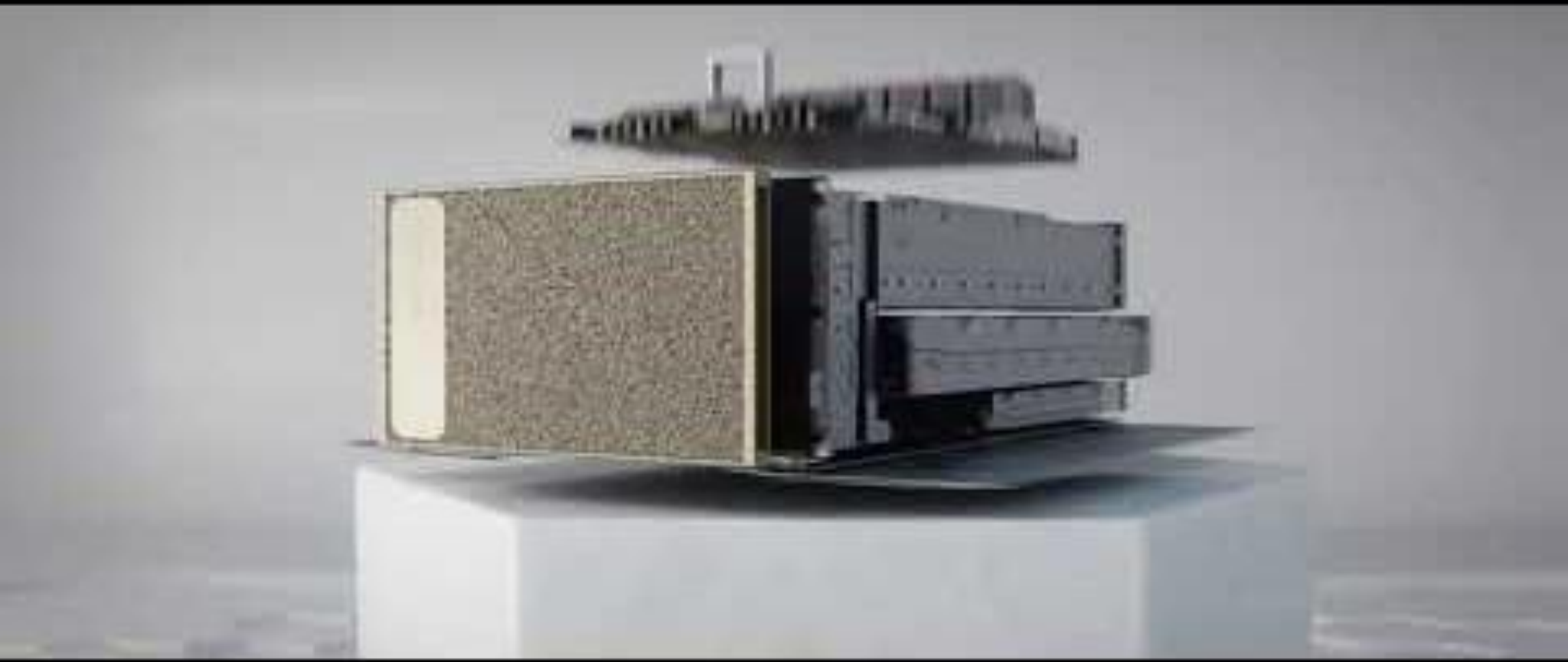
9X MORE PERFORMANCE IN 4 YEARS

Beyond Moore's Law With Full Stack Innovation



Geometric Mean of application speedups vs. P100 : Benchmark Application: Amber [PME-Cellulose_NVE], Chroma [szscl21_24_128], GROMACS [ADH Dodec], MILC [Apex Medium], NAMD [stmv_nve_cuda], PyTorch (BERT Large Fine Tuner], Quantum Espresso [AUSURF112-jR]; Random Forest FP32 [make_blobs (160000 x 64 : 10)], TensorFlow [ResNet-50], VASP 6 [Si Huge], |GPU node: with dual-socket CPUs with 4x P100, V100, or A100 GPUs.

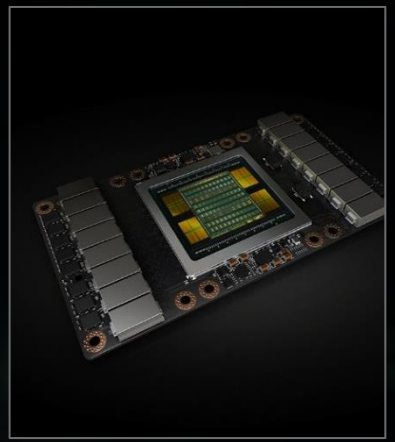
1. New Tensor Core
2. Strong Scaling
3. Elastic GPU
4. **Productivity**

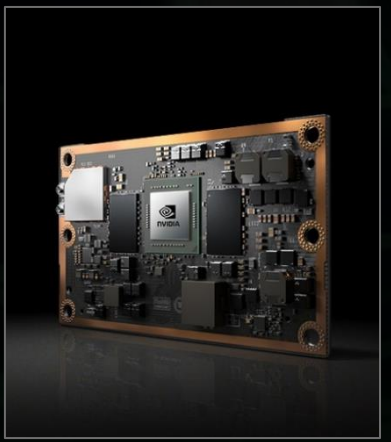


https://www.youtube.com/watch?v=TJcKYUTaBtg&t=3s

NVIDIA

One Platform. All challenges.



ONE ARCHITECTURE — CUDA

KEY ANNOUNCEMENT ASSETS

JHH Keynote On Demand

Tuesday, May 19th

Inside the NVIDIA Ampere Architecture
09:00 AM - 10:00 AM

CUDA New Features And Beyond
10:15 AM - 11:15 AM

CUDA on NVIDIA Ampere GPU Architecture: Taking Your Algorithms to the Next Level of Performance
11:30 AM - 12:30 PM

Inside the NVIDIA HPC SDK: the Compilers, Libraries and Tools for Accelerated Computing
1:30 PM - 2:30 PM

Wednesday, May 20th

Introducing NVIDIA DGX A100: The Universal AI System for Enterprise
9:00 AM - 10:00 AM

Accelerating Deep Learning Inference With Sparse Tensor Cores of Ampere GPU Architecture
1:30 PM - 2:30 PM

Mixed-Precision Training of Neural Networks
2:45 PM - 3:45 PM

Thursday, May 21st

Tensor Core Performance on NVIDIA GPUs: The Ultimate Guide
9:00 AM - 10:00 AM

Optimizing Applications for NVIDIA Ampere GPU Architecture
10:15 AM - 11:15 AM

Developing CUDA kernels to push Tensor Cores to the Absolute Limit on NVIDIA A100
11:30 AM - 12:30 PM

High-Performance Next-Generation Deep-Learning Clusters
1:30 PM - 2:30


NVIDIA