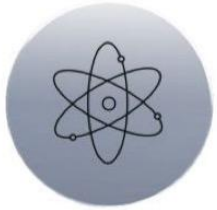
# NVIDIA Ampere GA102

Boyuan Tian, Vincent Wells, Katherine Yun

# Applications benefit from GPU

Graphics



Scientific Computing

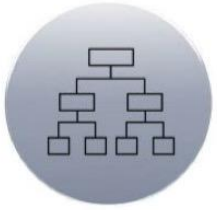
Data Analytics

AI Deep Learning Training

Edge AI Video Analytics

Cloud Gaming

Genomics

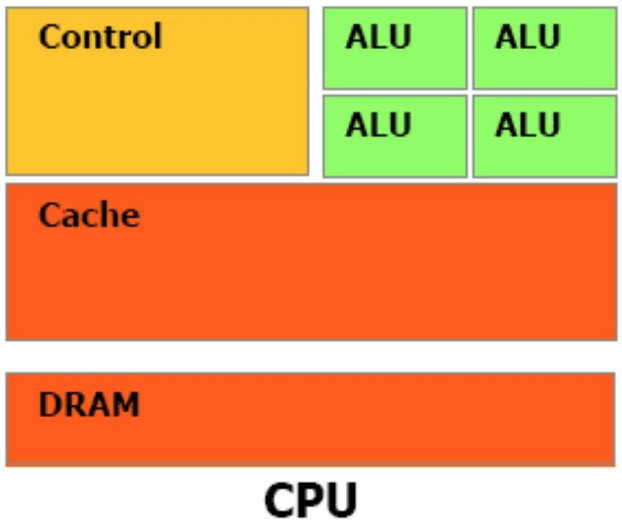


Classical Machine Learning

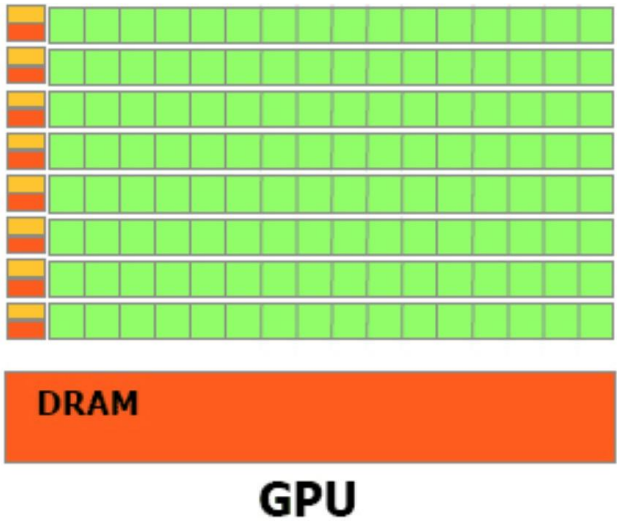
AI Deep Learning Inference

5G Private Networks

# GPU vs. CPU



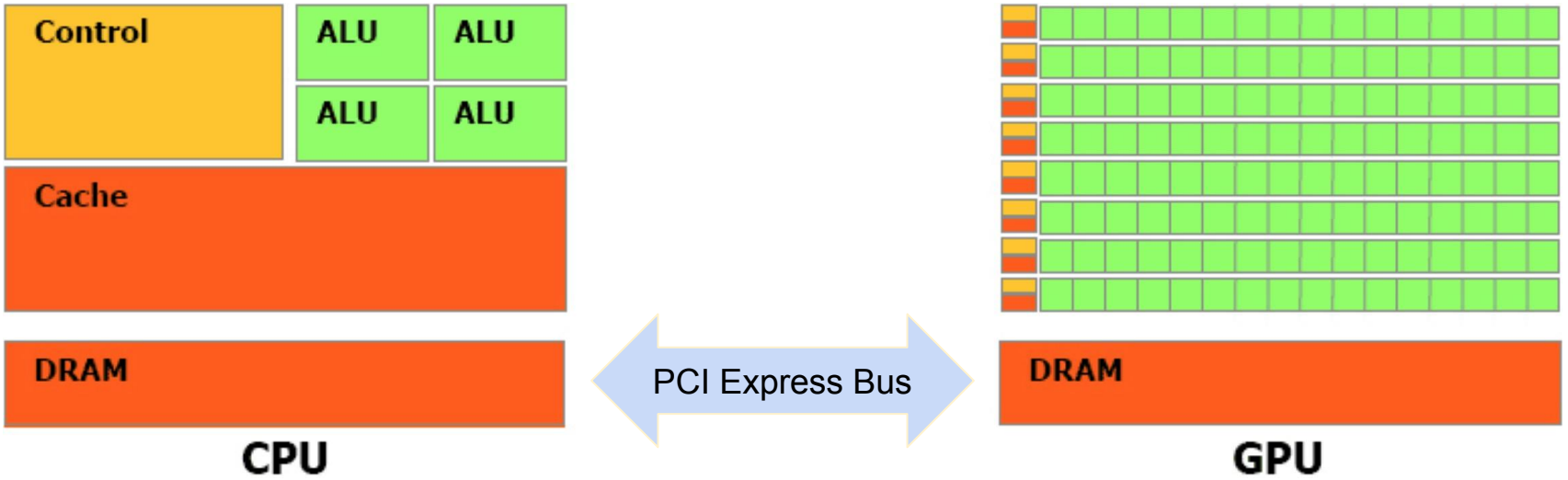

Latency-oriented




Throughput-oriented

*Credit : https://sites.google.com/site/daveshshingari/explorations/computer-architecture/gpu-architecture*

# Heterogeneous architecture



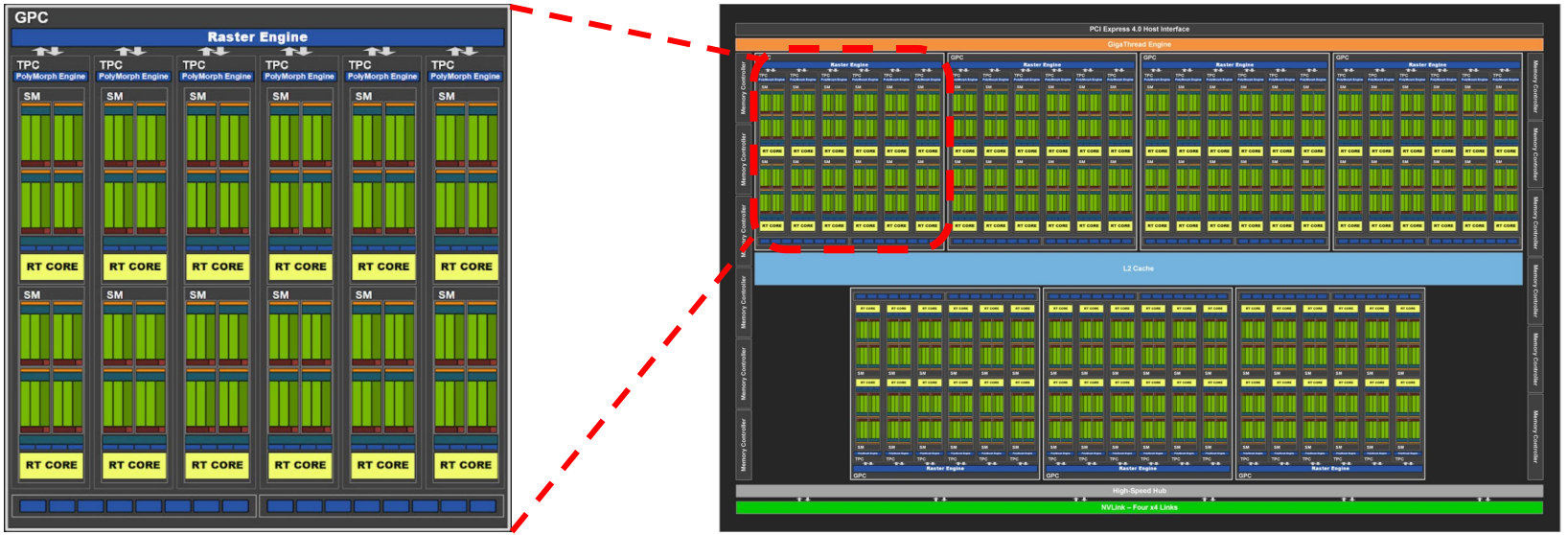

*Credit : https://sites.google.com/site/daveshshingari/explorations/computer-architecture/gpu-architecture*

# Overview

- 28.3 billion transistors
- Die area: 628.4 $mm^2$
- Samsung's 8nm

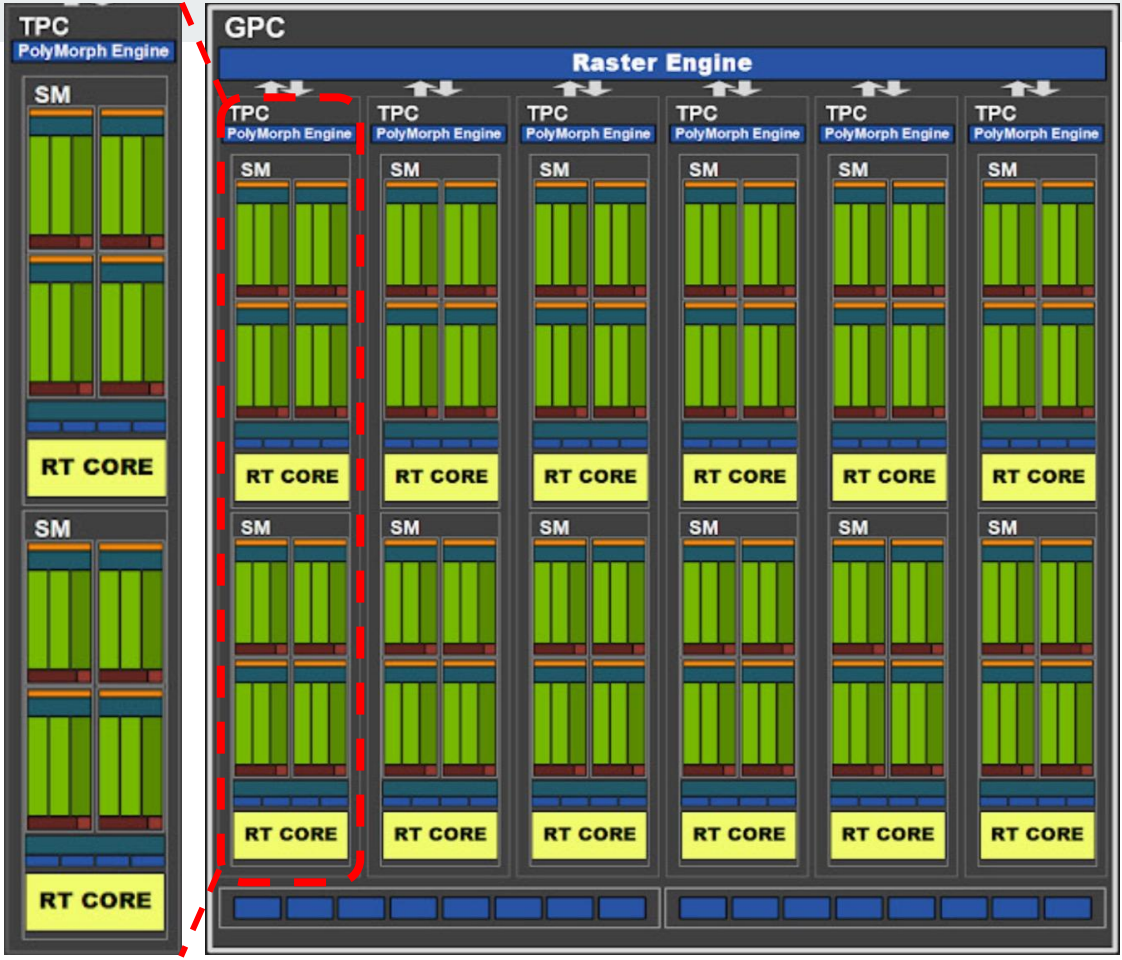
# Components

- Graphics Processing Clusters (GPCs)



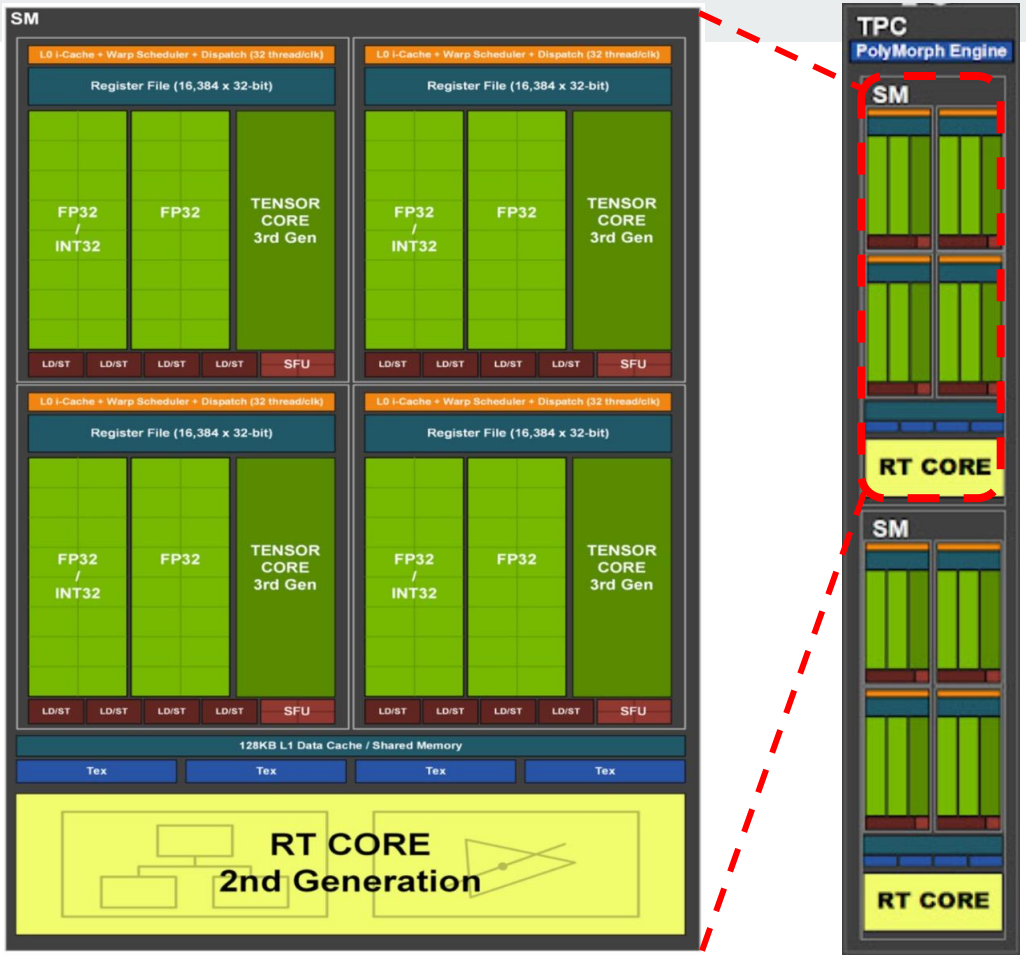
# Components

- Texture Processing Clusters (TPCs)
  - 6 per GPC
  - 41 per core



# Components

- Streaming Multiprocessor (SMs)
  - 2 per TPC
  - 82 per core

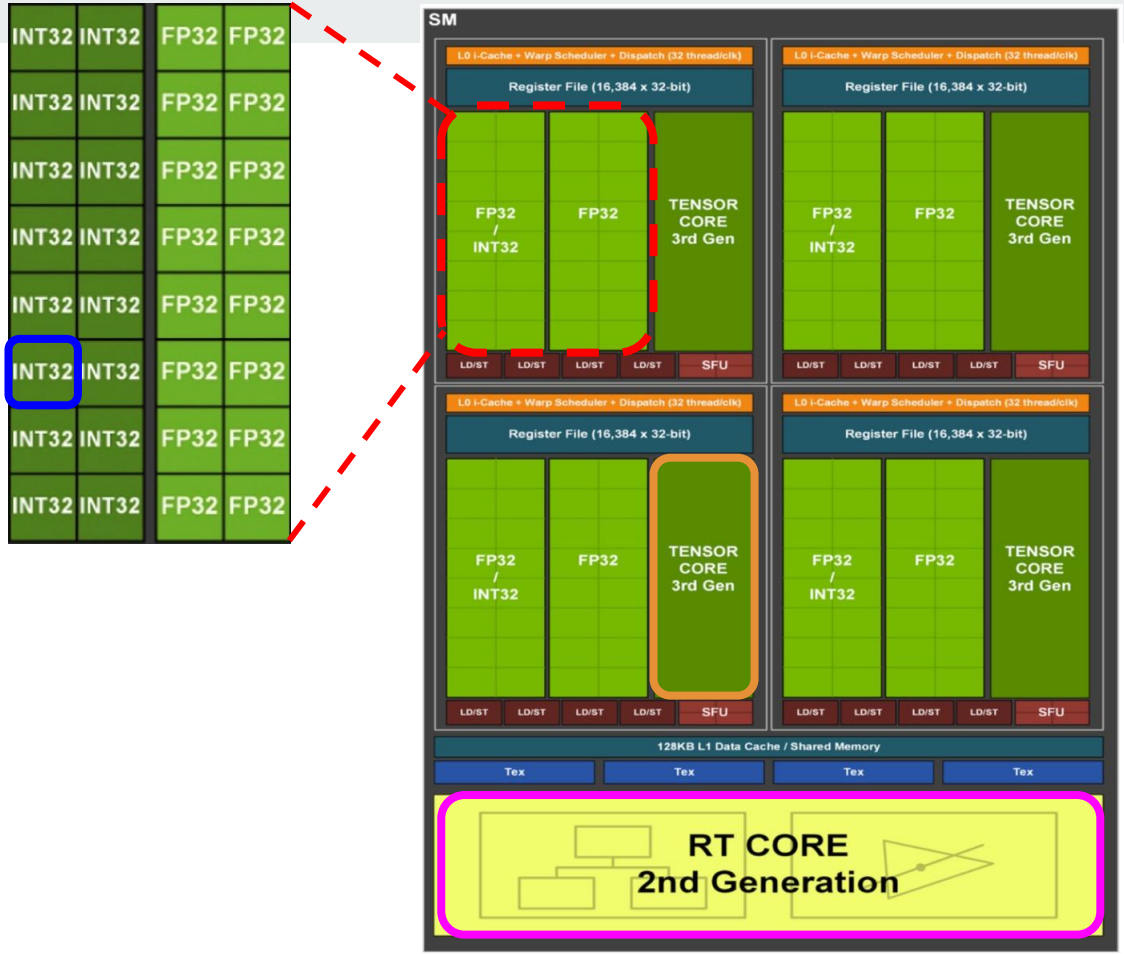


# Components

- Shared resources
  - L0 i-Cache
  - Warp scheduler
  - Dispatch
  - Register files

# Components

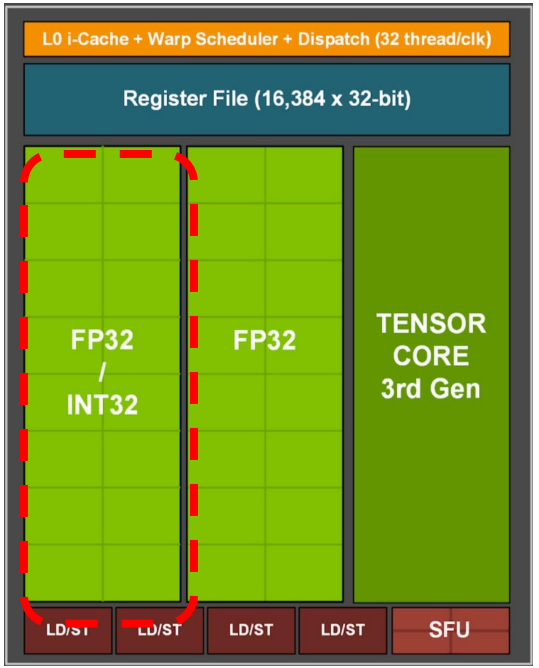
- CUDA cores
  - Arithmetic operation
  - 128 per SM
  - 10496 per core
- Tensor cores
  - Matrix-matrix multiplication
  - 4 per SM
  - 328 per core
- RT cores
  - Ray tracing
  - 1 per SM
  - 82 per core



# CUDA core

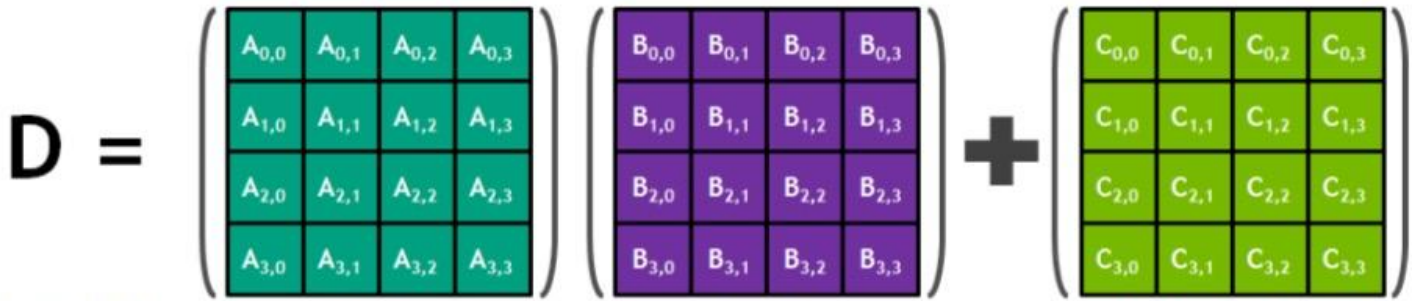
- Fully pipelined ALUs and FPUs

- Ampere
  - 64 INT32 / FP32 + 64 FP32 / SM
- Volta, Turing
  - 64 INT32 + 64 FP32 / SM

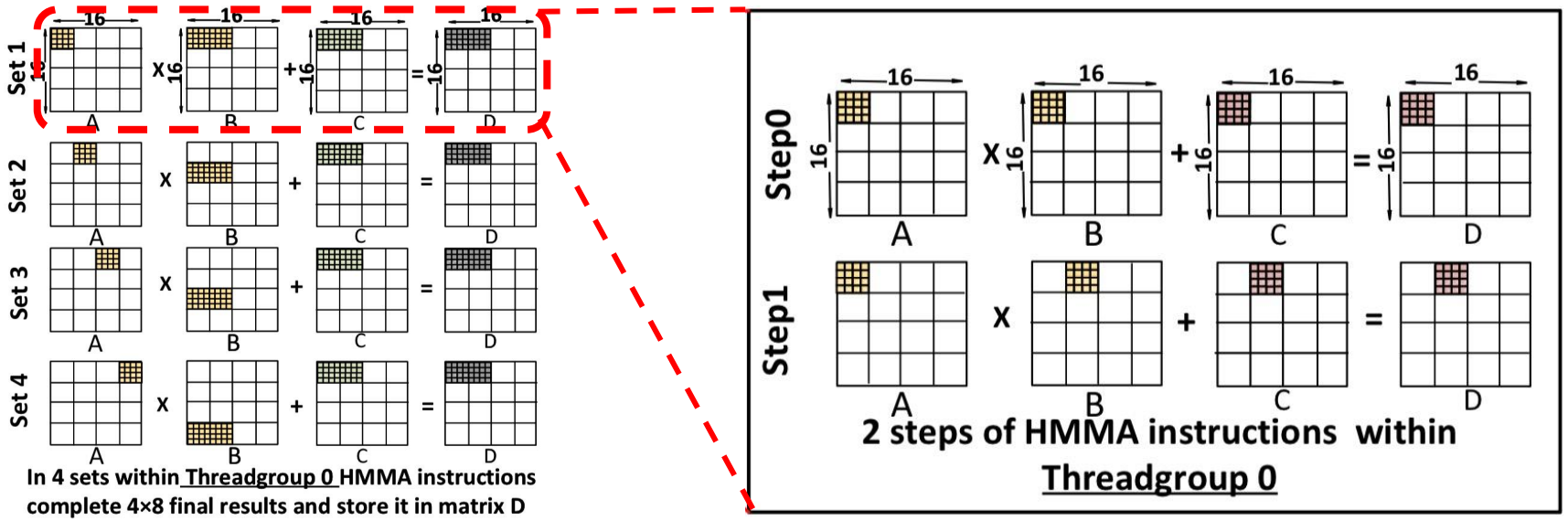


# Tensor core

- 4 x 4 matrix multiplication
- Multiply-Accumulate Operation (MAC):
  - 128 in total = 64 multiplications + 64 accumulations



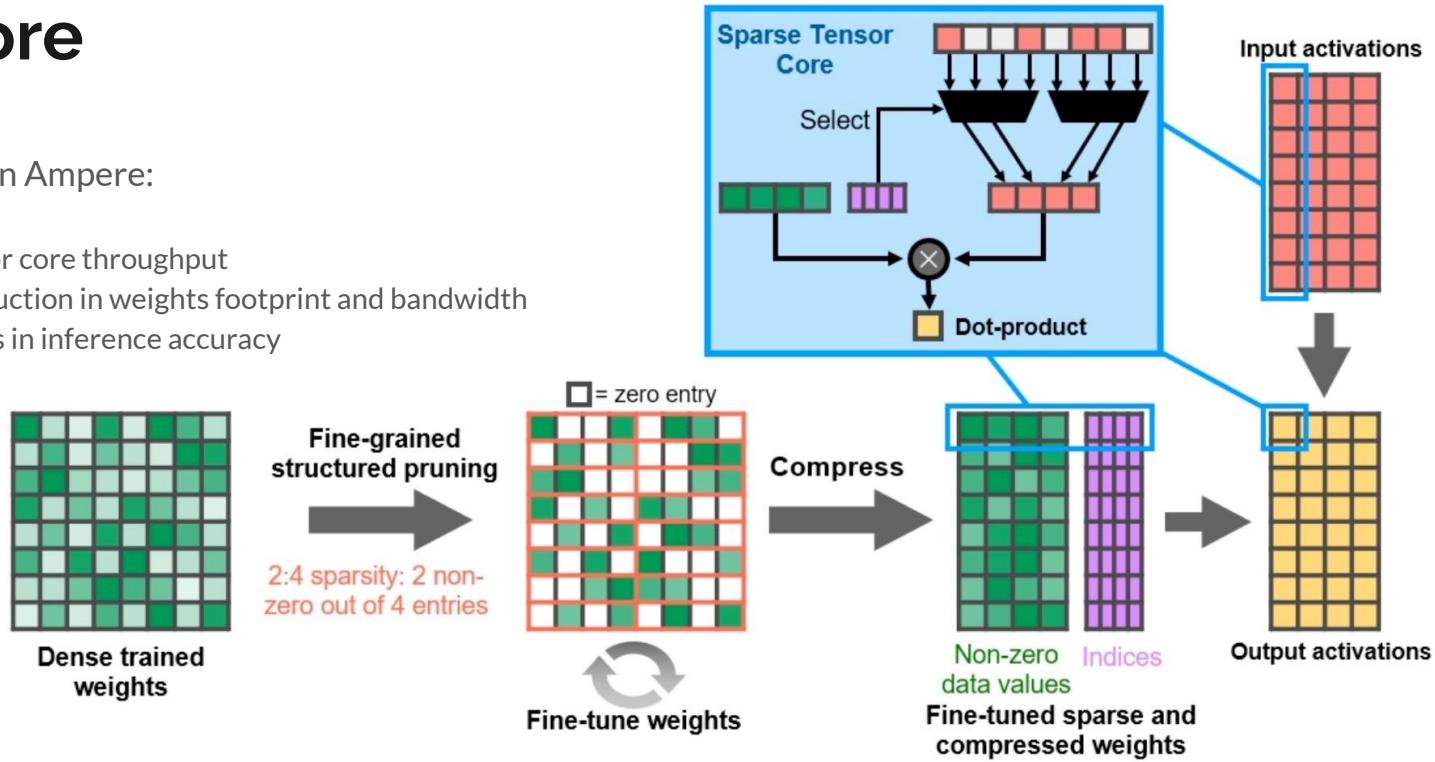
# Tensor core




*Credit : Raihan, Md Aamir, Negar Goli, and Tor M. Aamodt. "Modeling deep learning accelerator enabled GPUs." 2019 ISPASS*

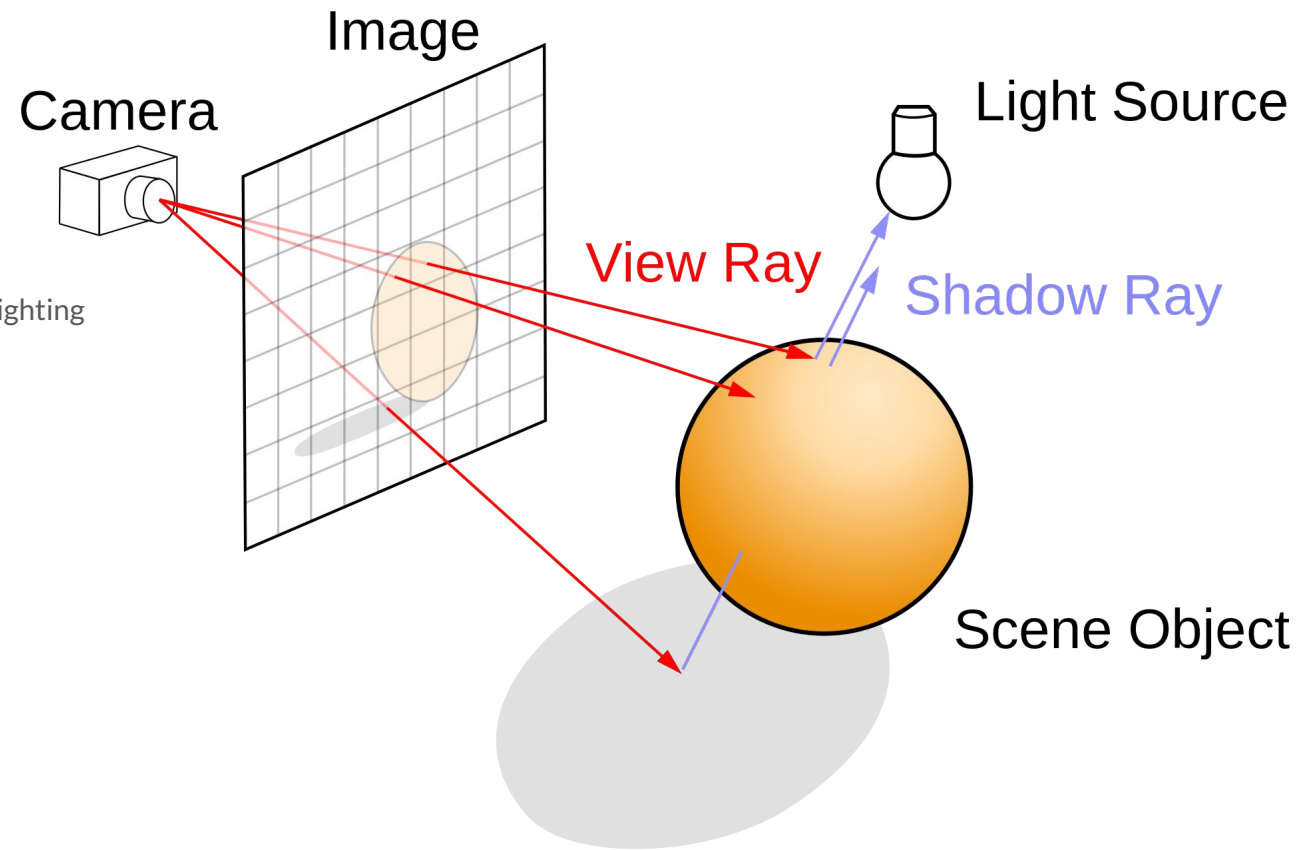
# Tensor core

- New feature in Ampere:
  - Sparsity
  - 2x Tensor core throughput
  - ~ 2x reduction in weights footprint and bandwidth
  - ~ No loss in inference accuracy



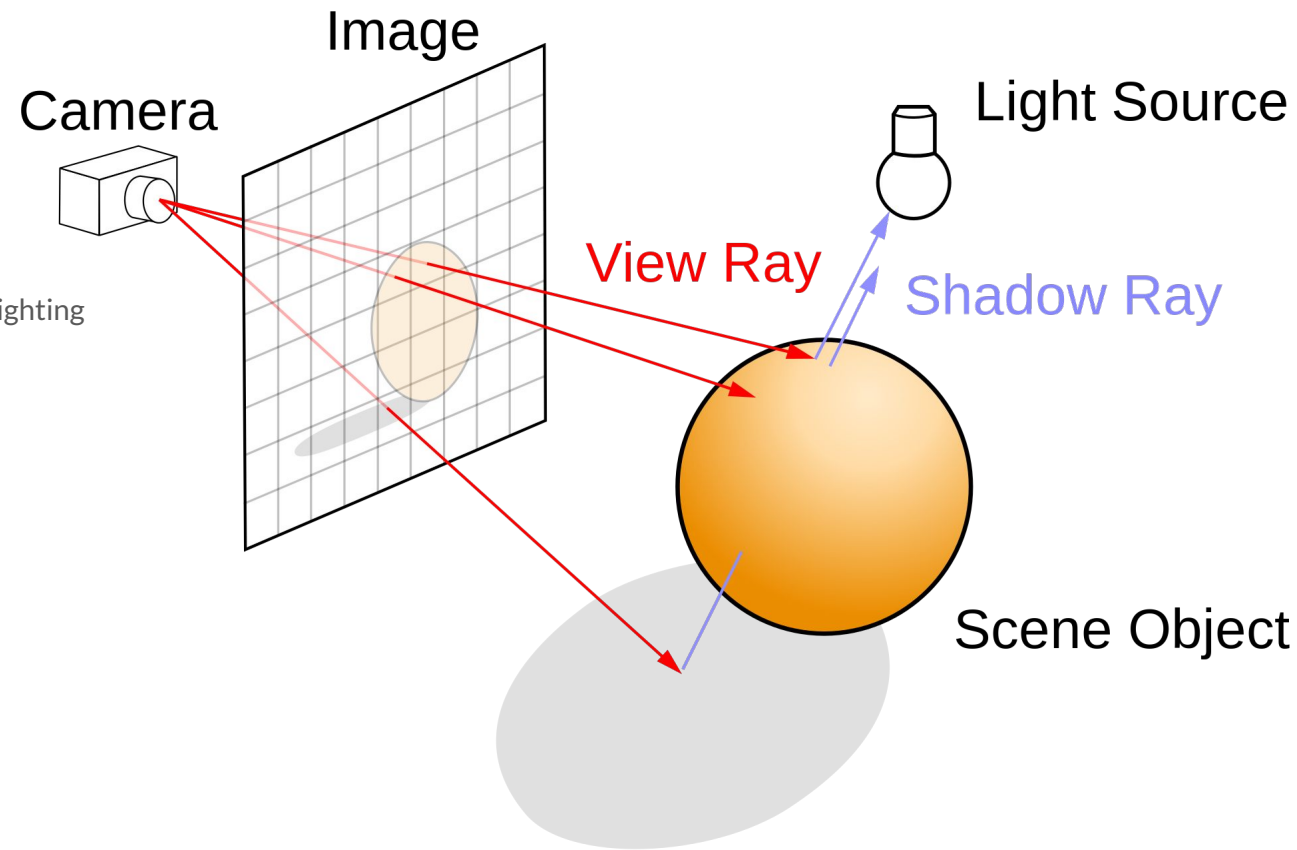
# RT core

- Ray tracing:
  - Realistic simulate lighting
  - Physically correct



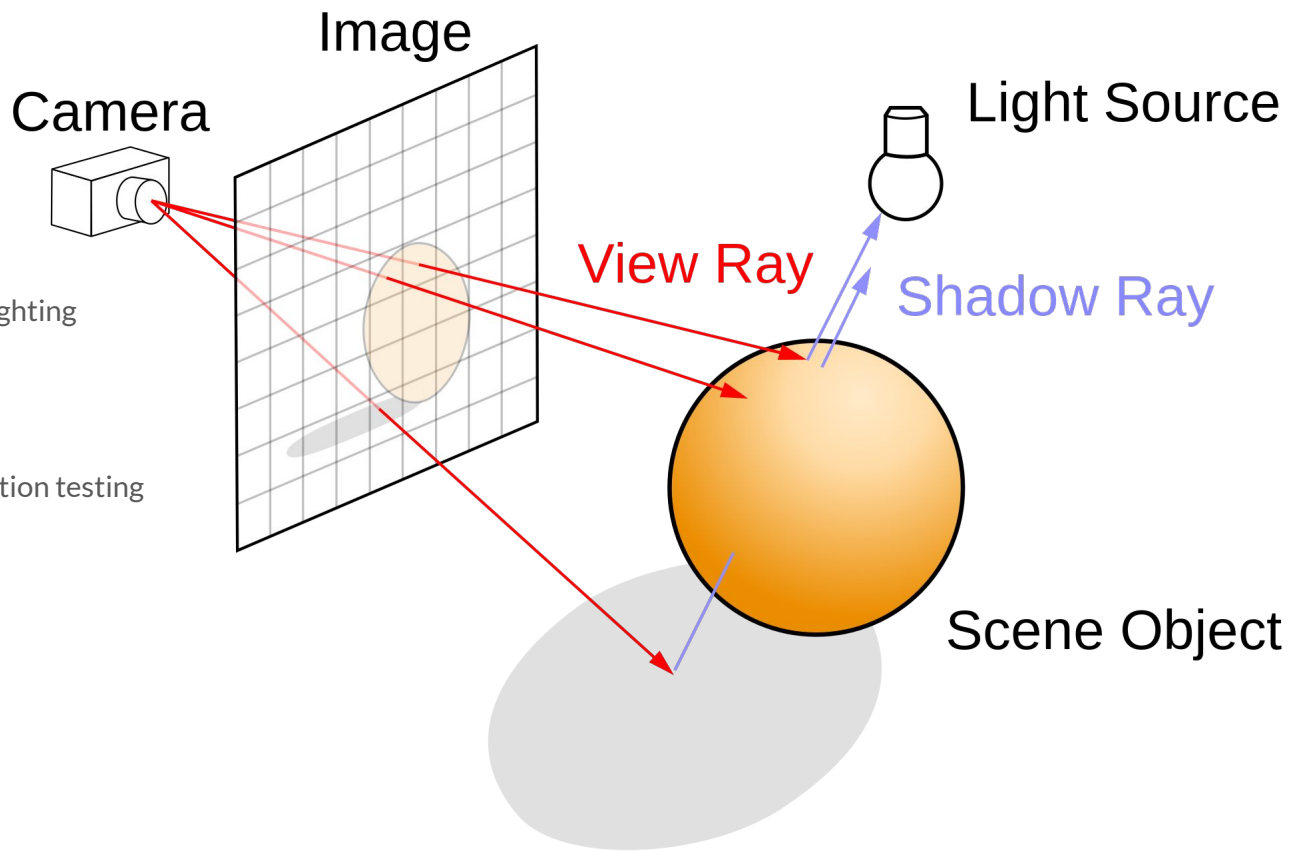
# RT core

- Ray tracing:
  - Realistic simulate lighting
  - Physically correct
- Basic ray tracing



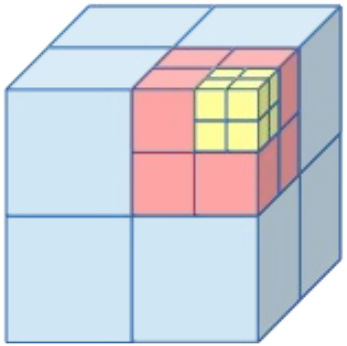
# RT core

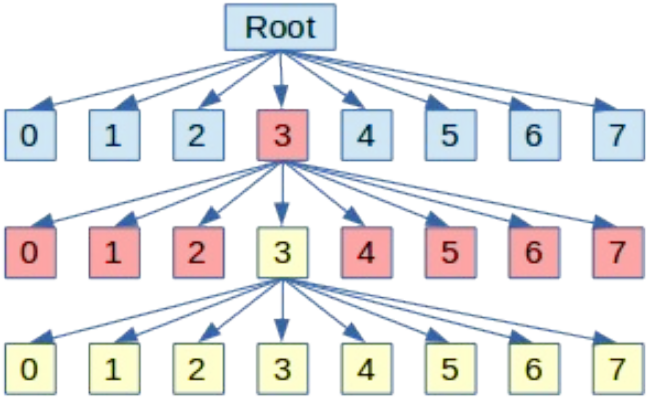
- Ray tracing:
  - Realistic simulate lighting
  - Physically correct
- Basic ray tracing
- Optimizations
  - Accelerate intersection testing



# RT core

- Ray tracing:
  - Realistic simulate lighting
  - Physically correct
- Basic ray tracing
- Optimizations
  - Accelerate intersection testing
  - Reduce the mesh search cost

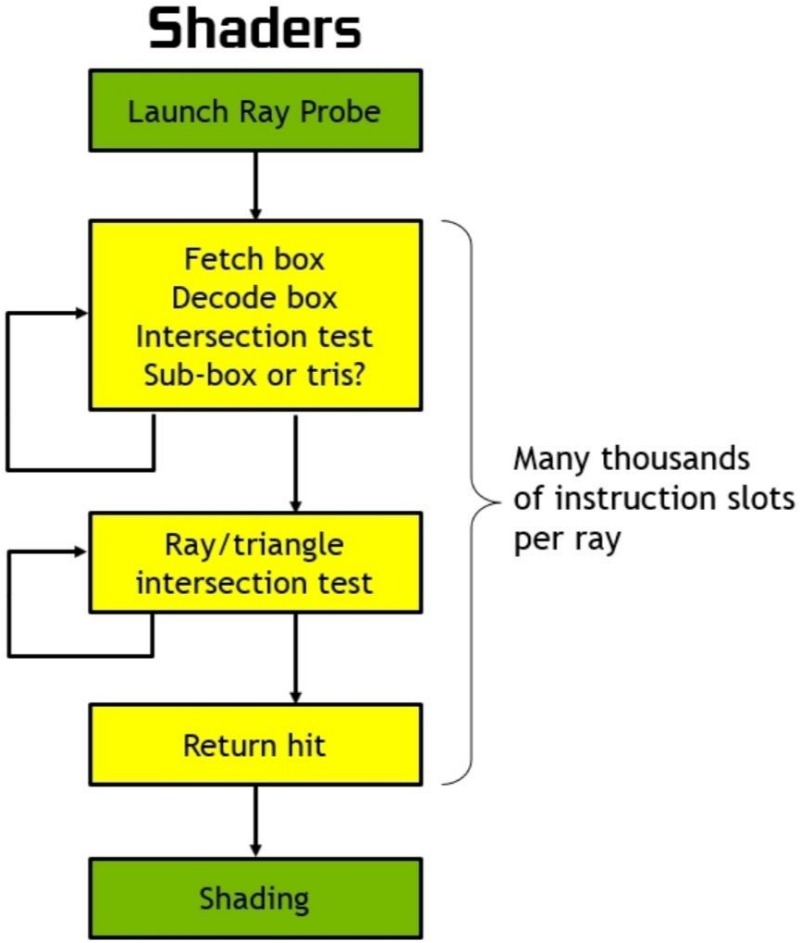





*Credit : https://geidav.wordpress.com/2014/07/18/advanced-octrees-1-preliminaries-insertion-strategies-and-max-tree-depth/*

# RT core

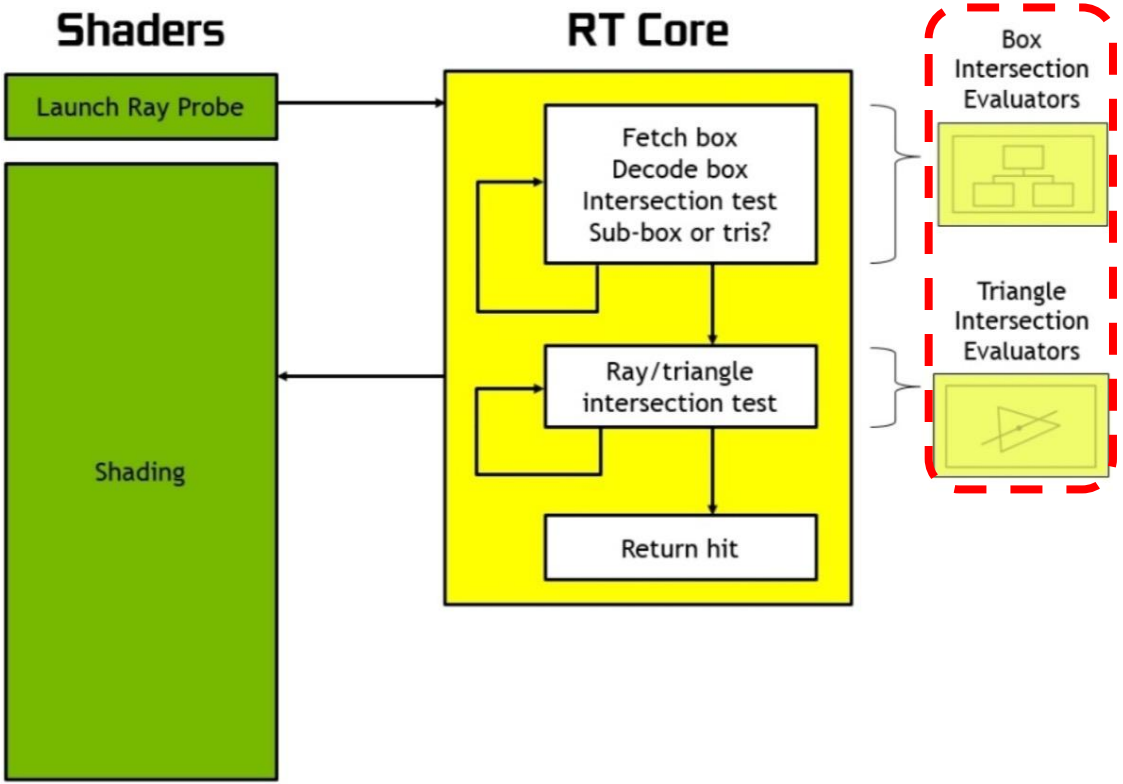
- Ray tracing with CUDA cores



# RT core

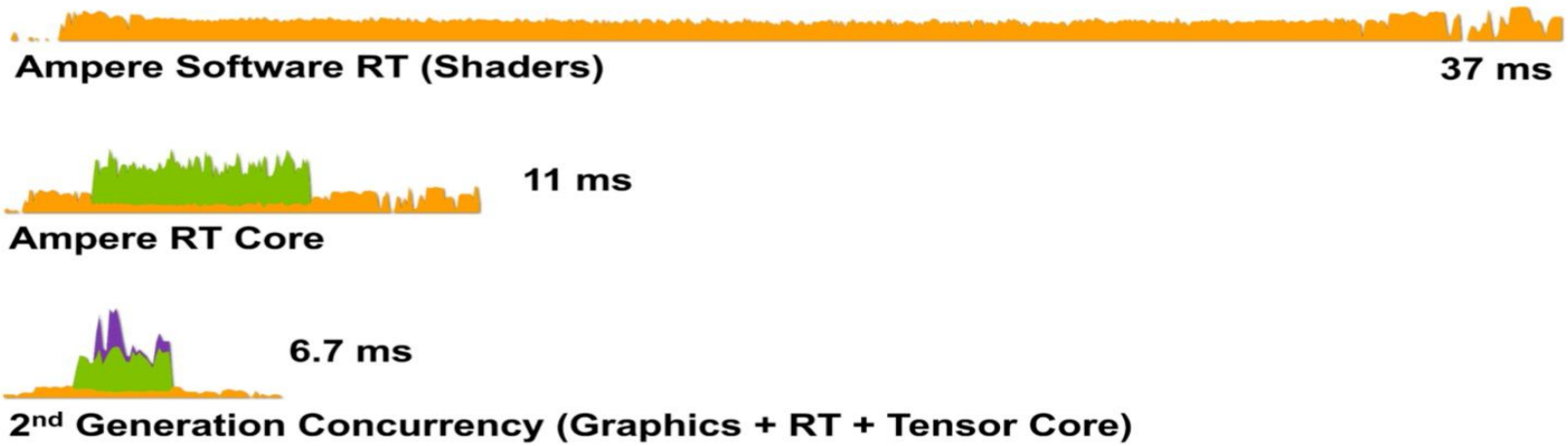
- Ray tracing with RT cores

- Dedicated hardwares
  - Box intersection checking
  - Triangle intersection checking



# RT core

- New features on Ampere
  - Concurrency on RT core and Tensor core



# Memory Hierarchy

- 7 Graphics Processing Clusters (GPCs)
  - L2 Cache (6133 KB)
  - 12 32-bit memory controllers
    - Each paired with 512KB of L2 cache
- 84 Streaming Multiprocessors (SMs)
  - Combined L1 data cache/shared memory (128KB)
    - Increased by 33% compared to Turing
    - Configurable based on compute workloads
  - Each SM has 4 processing blocks (partitions)
    - Register file (64KB)
    - L0 instruction cache

# L1 Data Cache/Shared Memory

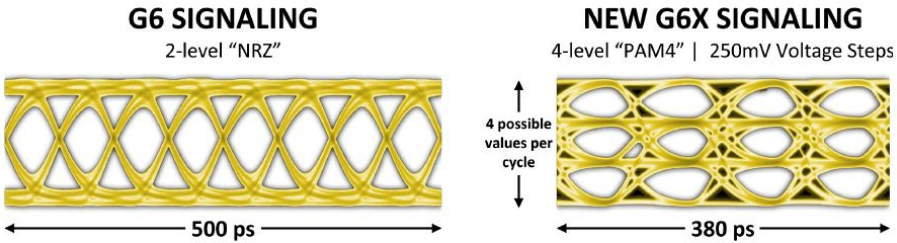
- SM level memory
  - Accessible by threads within a SM
- Unified architecture for shared memory, L1 data cache, and texture caching
- Workload-based reconfiguration
  - Up to 128 KB per SM

# L1 Data Cache/Shared Memory Cont'd

- Configuration supported (compute mode)
  - 128 KB L1 + 0 KB Shared Memory
  - ……
  - 64 KB L1 + 64 KB Shared Memory
  - 28 KB L1 + 100 KB Shared Memory
- Graphics workloads and async compute
  - 64 KB  L1 data/texture cache (32 KB on Turing)
  - 48 KB shared memory
- Features double shared memory bandwidth
  - 128 bytes/clock/SM (doubled compared to Turing)

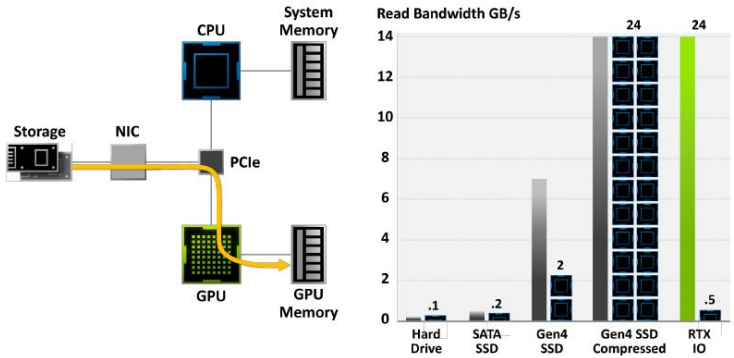
# GDDR6X Memory

- New to Ampere family processors
  - Based on prior GDDR6 memory standard in 2018
- Peak memory bandwidth of 936 GB/sec with PAM4 signaling
  - Double I/O data transfer  rate
  - Sends two bits on each clock edge (rising and falling edges)
  - Voltage levels are divided into 250 mV steps
    - 00, 01, 10, 11 (DDR technology)

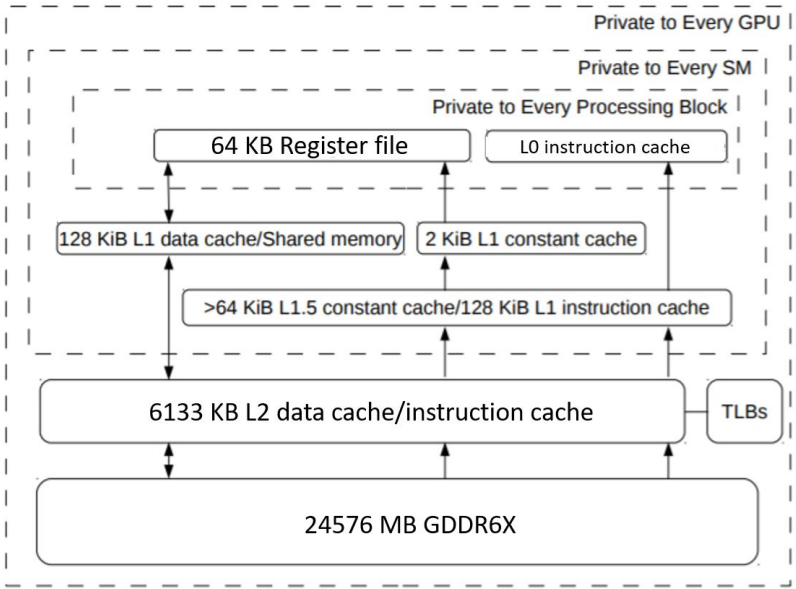


# RTX IO

- Gen4 SSDs with up to 7GB/sec read bandwidth
- CPU file systems become a bottleneck in loading game memory data
- GPU-based lossless decompression
    - Reads remain compressed data and delivers to GPU for decompression
    - Removes decompression load from the CPU to GPU



# Memory Hierarchy Overview



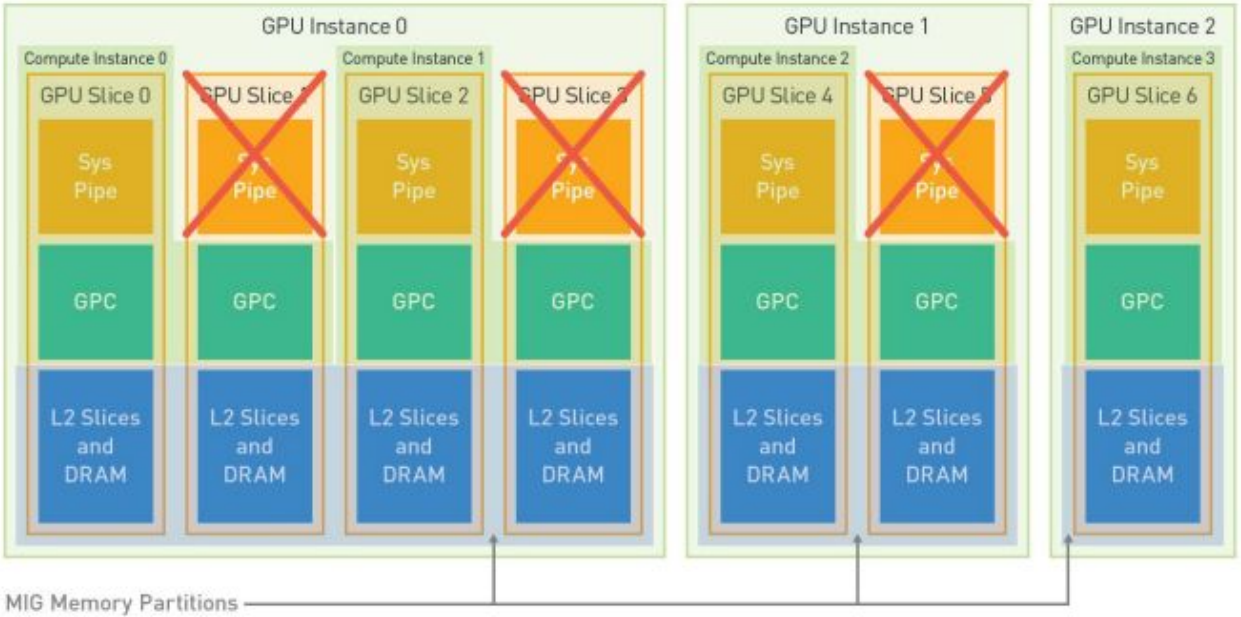
# Parallelism Support

- CUDA Taskgraphs
  - Dependency graph of GPU operations
  - Enable a “define-once/run-repeatedly” execution flow
  - Generally many independent operations to run in parallel on the available cores
  - A100 adds hardware features to accelerate traversing a task graph
- Can use MIG to divide a GPU into GPU instances and run in parallel

# Multi-Instance GPU

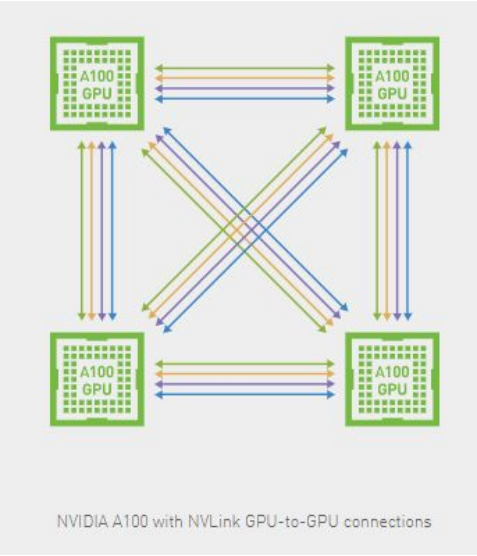
- Multi-Instance GPU (MIG)
  - New feature which allows the GPU to partitioned into as many as 7 separate CUDA GPU instances
  - Each instance has its own path through the entire memory system (on-chip crossbar ports, L2 cache banks, mem. Controllers, DRAM address buses)
  - Especially useful for Cloud Service Providers

# Multi-Instance GPU Example



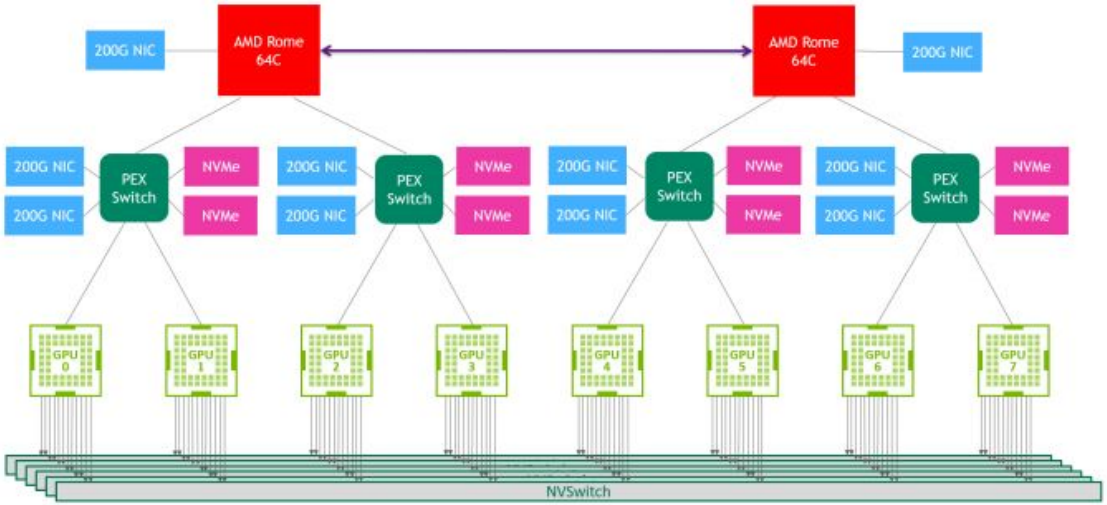
# Multi-GPU

- 3rd Generation NVLink
  - Interconnect multiple GPUs on a node using NVSwitch
    - ~2x faster than previous generation
  - Allows for up to 600 Gb/sec total bandwidth out of 12 links on a given A100 GPU
    - ~10X faster t han PCIe Gen4



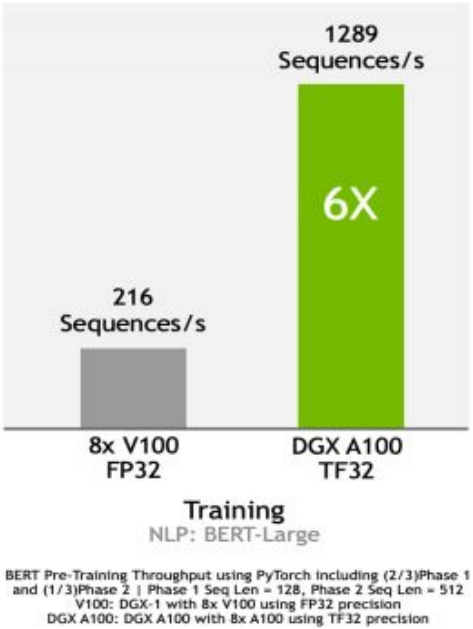
NVIDIA A100 with NVLink GPU-to-GPU connections

# Multi-GPU Example

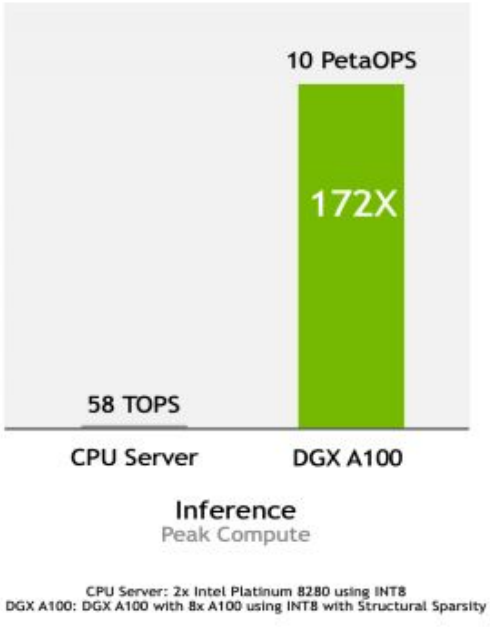



Note Third-Generation NVLink connectivity through NVSwitches.

# Multi-GPU Performance




BERT Pre-Training Throughput using PyTorch including (2/3)Phase 1 and (1/3)Phase 2 | Phase 1 Seq Len = 128, Phase 2 Seq Len = 512
V100: DGX-1 with 8x V100 using FP32 precision
DGX A100: DGX A100 with 8x A100 using TF32 precision




CPU Server: 2x Intel Platinum 8280 using INT8
DGX A100: DGX A100 with 8x A100 using INT8 with Structural Sparsity

# Multi-Node Parallelism

- NVIDIA Magnum IO & Mellanox Interconnect Solutions
  - Full support for NVIDIA Magnum IO APIs, which accelerate multi-GPU, multi-node systems to maximize IO performance
  - Compatible with Mellanox Infiniband and Ethernet connections
  - Supports PCIe Gen 4 with SR-IOV which allows it support faster network interfaces cards like 200 Gbit/sec Mellanox ConnectX-6 VPI HDR Infiniband

# Any questions?

# References

“NVIDIA A100 Tensor Core GPU Architecture.” NVIDIA, 2020.

J. Choquette and W. Gandhi, "NVIDIA A100 GPU: Performance & Innovation for GPU Computing," 2020 IEEE Hot Chips 32 Symposium (HCS), Palo Alto, CA, USA, 2020, pp. 1-43, doi: 10.1109/HCS49909.2020.9220622.

Y. Tsai, T. Cojean, and H. Anzt, “Evaluating the Performance of NVIDIA's A100 Ampere GPU for Sparse Linear Algebra Computations,” ArXiv, 2020, abs/2008.08478.

# References


“NVIDIA Ampere GA102 GPU Architecture.” NVIDIA, 2020.

“NVIDIA Turing GPU Architecture.” NVIDIA, 2018.

“CUDA C++ Programming Guide”. NVIDIA, 2020

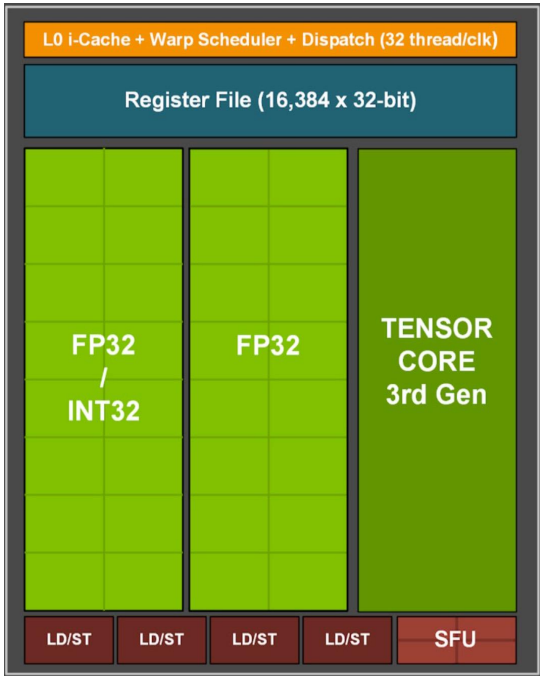
Raihan, Md Aamir, Negar Goli, and Tor M. Aamodt. "Modeling deep learning accelerator enabled GPUs." 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2019.

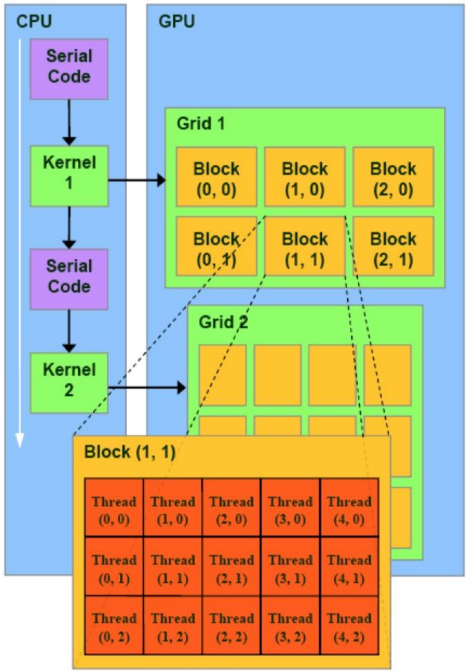
EECS 573 Microarchitecture Data Parallel Architectures: GPUs, Todd Austin, University of Michigan, 2014

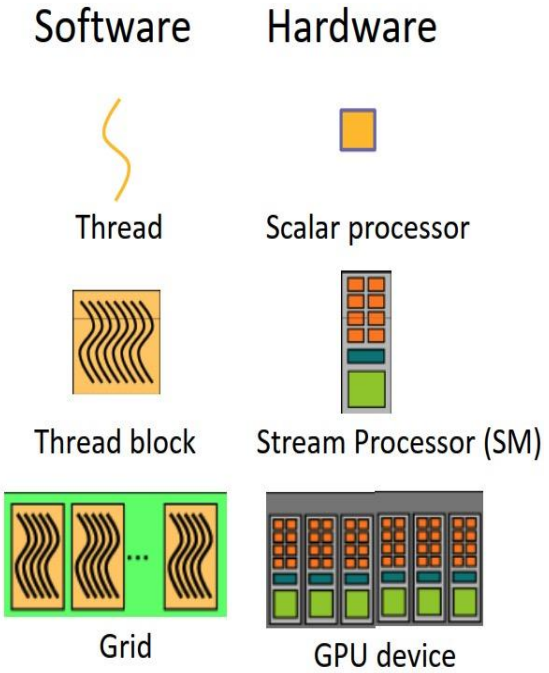
Advanced Computer Architecture, Data-Level Parallel Architectures: GPUs, Paul Kelly, ICL, 2019

# Backup slides

# SM deep dive

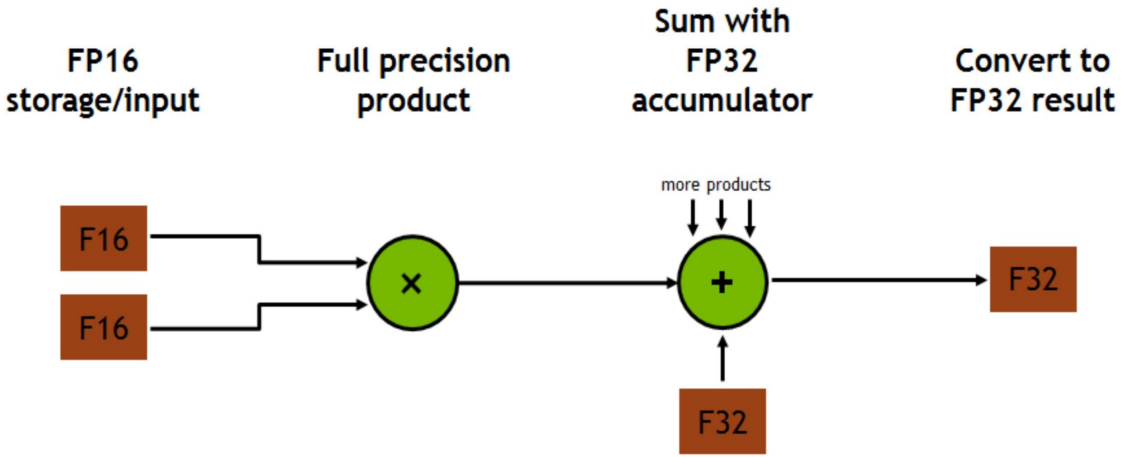

L0 i-Cache + Warp Scheduler + Dispatch (32 thread/clk)
Register File (16,384 x 32-bit)
FP32 / INT32
FP32
TENSOR CORE 3rd Gen
LD/ST
LD/ST
LD/ST
LD/ST
SFU


CPU
GPU
Serial Code
Kernel 1
Serial Code
Kernel 2
Grid 1
Block (0, 0)
Block (1, 0)
Block (2, 0)
Block (0, 1)
Block (1, 1)
Block (2, 1)
Grid 2
Block (1, 1)
Thread (0, 0)
Thread (1, 0)
Thread (2, 0)
Thread (3, 0)
Thread (4, 0)
Thread (0, 1)
Thread (1, 1)
Thread (2, 1)
Thread (3, 1)
Thread (4, 1)
Thread (0, 2)
Thread (1, 2)
Thread (2, 2)
Thread (3, 2)
Thread (4, 2)


Software
Hardware
Thread
Scalar processor
Thread block
Stream Processor (SM)
Grid
GPU device

# Tensor core

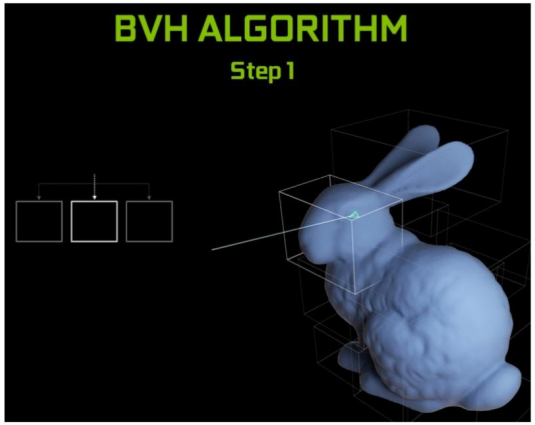
- Mixed-precision Operation



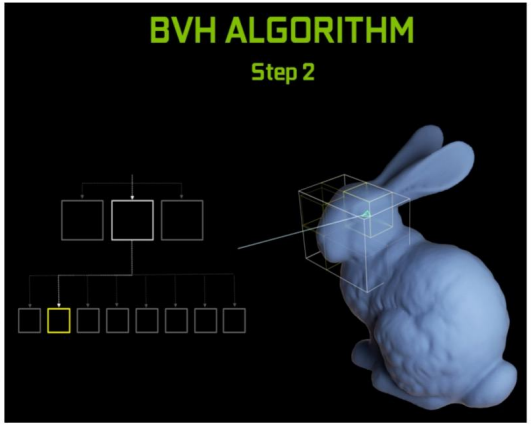

| Precision | Throughput (TOPS) |
|---|---|
| FP 16 | 144 |
| INT 8 | 288 |
| INT 4 | 455 |

# RT core

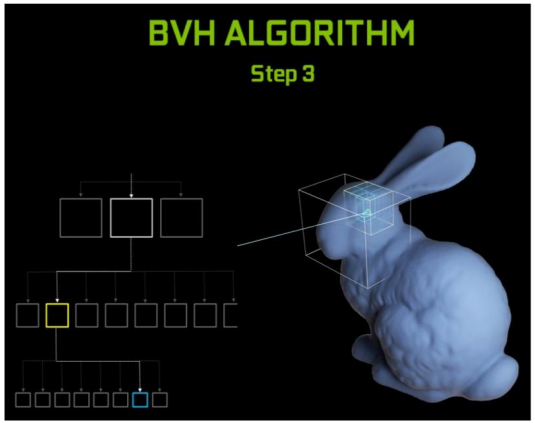
- Ray tracing:
  - Realistic simulate lighting
  - Physically correct
- Basic ray tracing
- Optimizations
  - Accelerate intersection testing
  - Reduce the number of rays
    - Bounding volume hierarchy



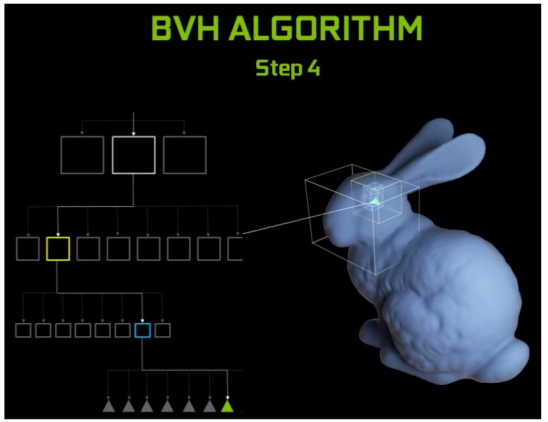

Step 1




Step 2



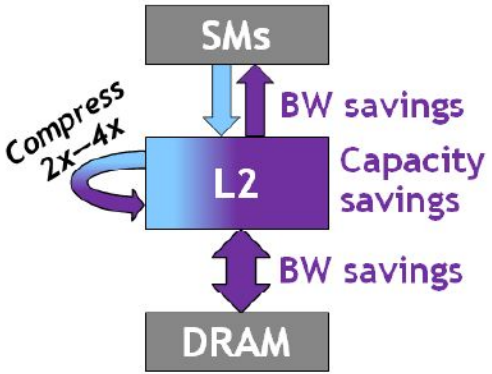

Step 3




Step 4

# A100 L2 Cache Memory

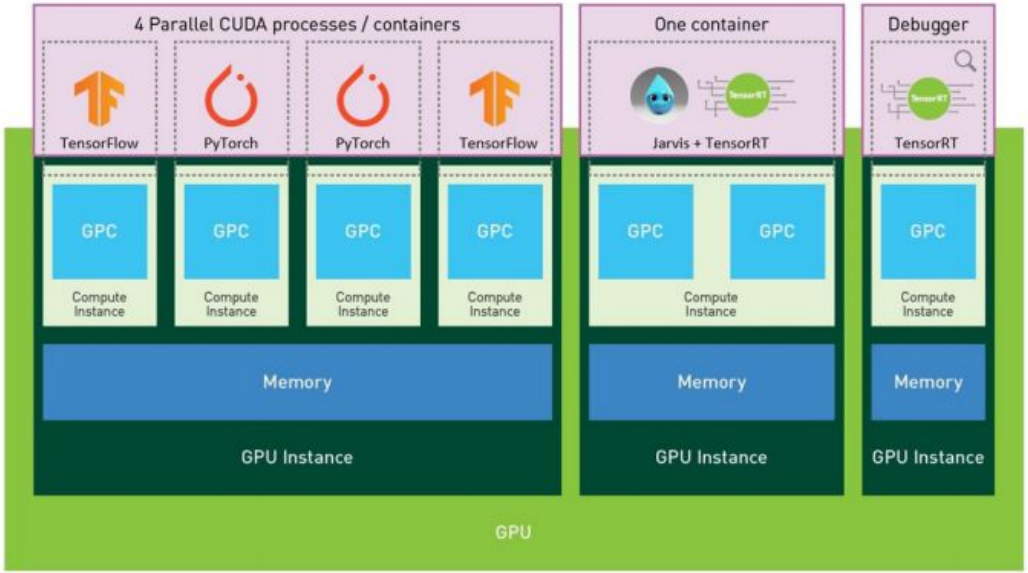
- A100 Tensor Core includes 40 MB of L2 cache
  - 6.7x larger than Tesla V100 L2 Cache
  - L2 cache is divided into two partitions to enable higher bandwidth
    - Each is divided into 40 L2 cache slices
    - 8 512 KB L2 slices are associated with each memory controller
- Compute Data Compression
  - Saves up to 4x DRAM read/write bandwidth,
  - Saves up to 4x L2 read bandwidth, and up to 2x L2 capacity.



# A100 HBM2 DRAM Subsystem

- Memory stacks located on the same physical package as the GPU
- A100 GPU includes 40 GB of fast HBM2 DRAM
- HBM2 delivers 1555 GB/sec memory bandwidth
  - With 1215 MHz (DDR) data rate
  - 1.7 higher than Tesla V100
- Error Correction Code (ECC)
  - Provides higher reliability for compute applications that are sensitive to data corruption
  - Important in large-scale cluster environments

# Parallelism Support -- Multi-Instance GPU




Example of multiple independent GPU Compute workloads running in parallel using a MIG configuration on an A100 GPU with three GPU Instances and variable numbers of Compute Instances within each GPU Instance.