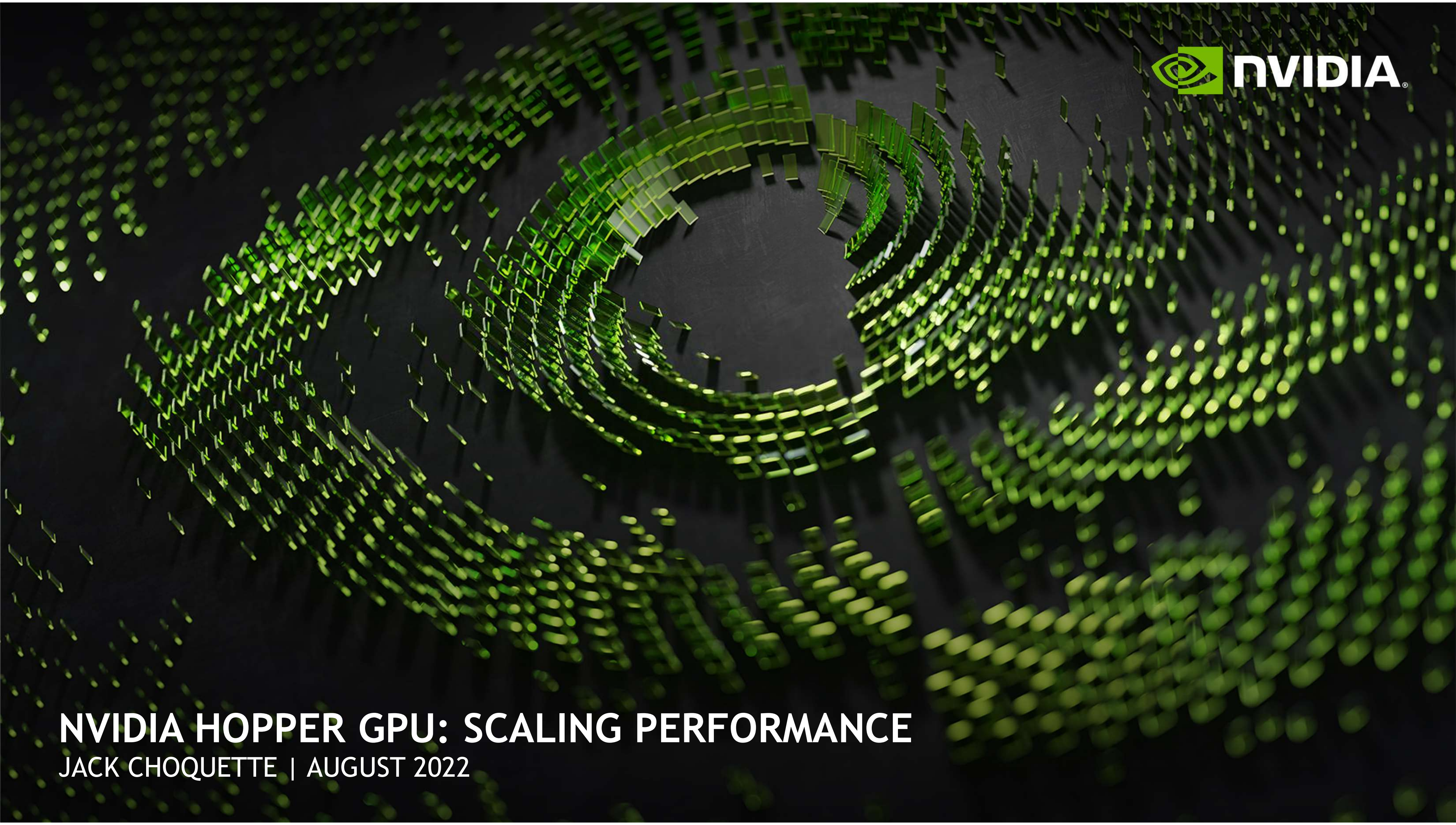
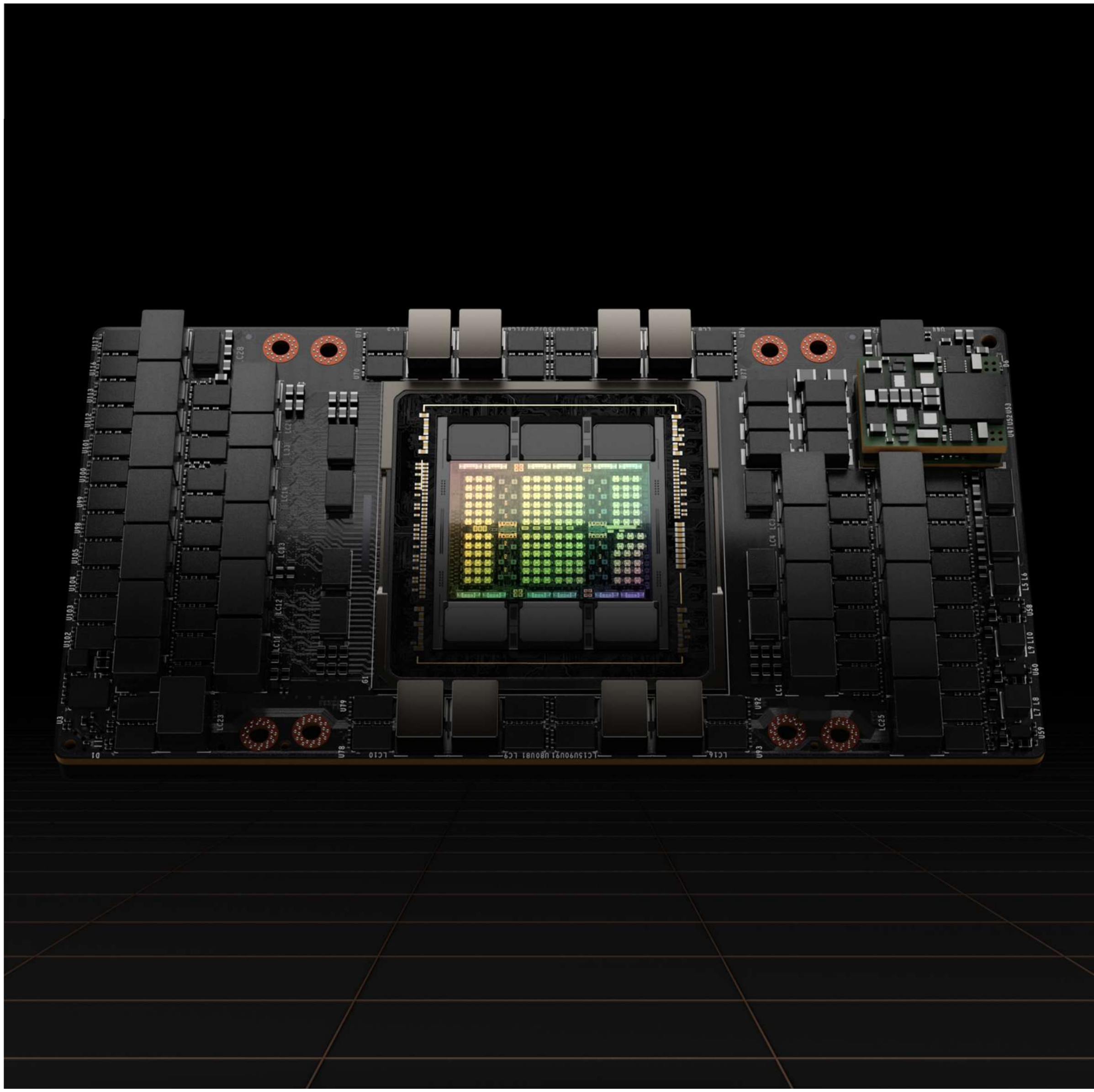


NVIDIA HOPPER GPU: SCALING PERFORMANCE

JACK CHOQUETTE | AUGUST 2022

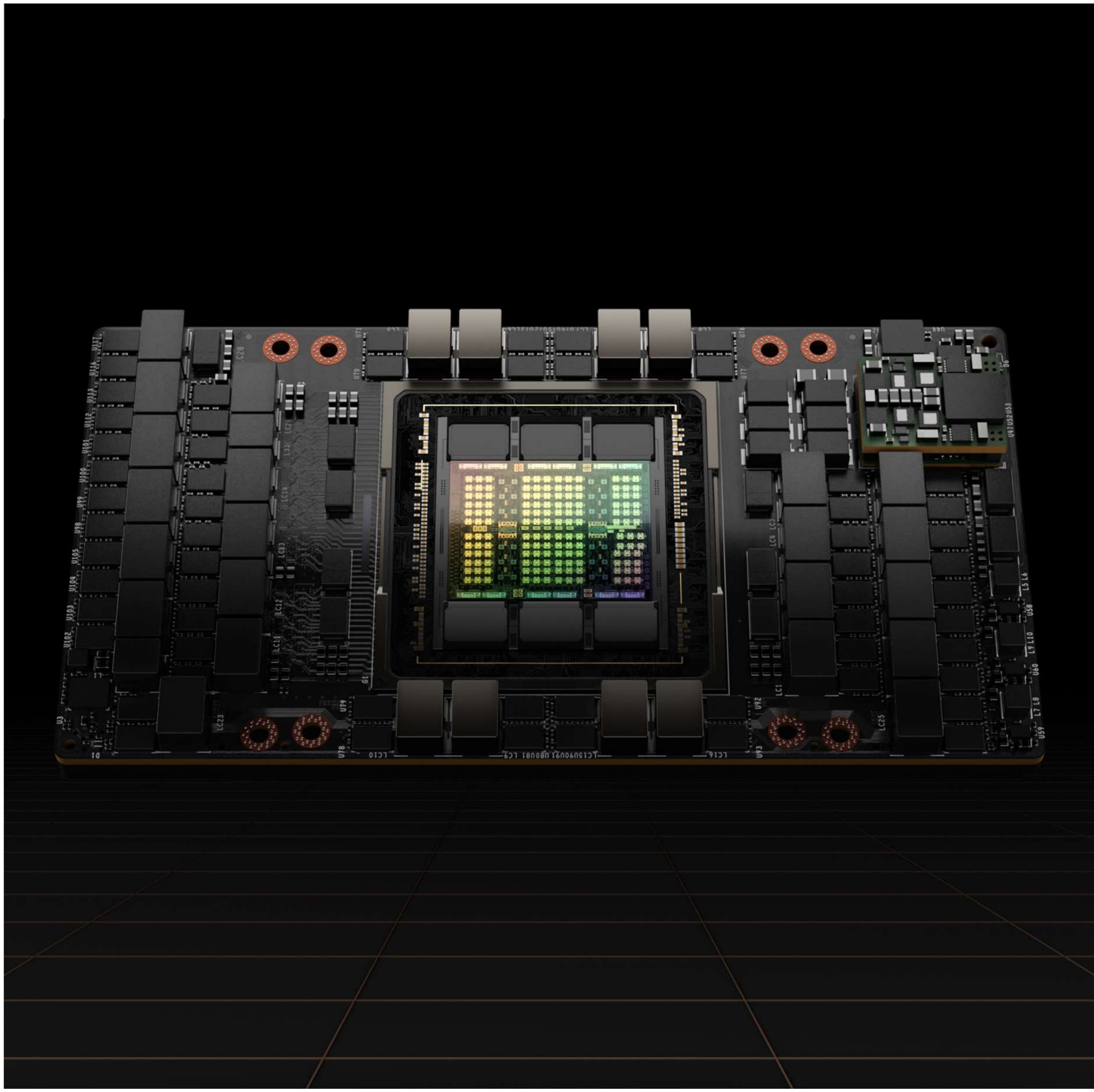
AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



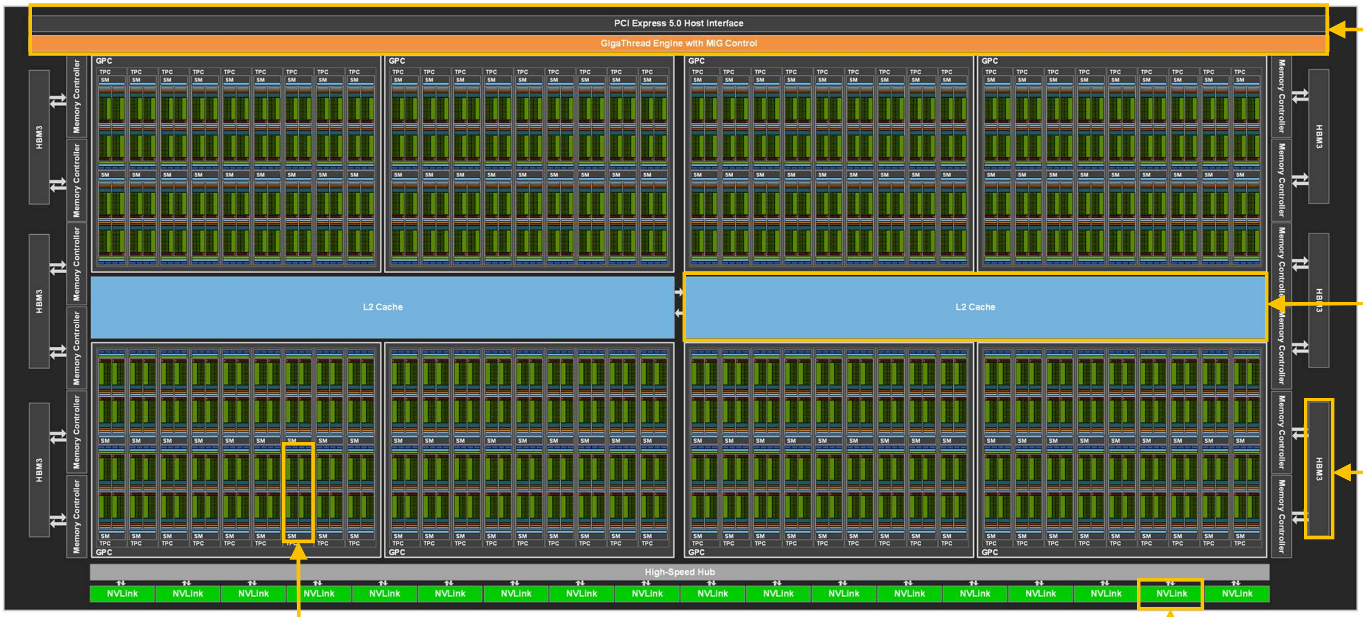
AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



HOPPER H100 TENSOR CORE GPU

80B Transistors, TSMC 4N



2nd Gen Multi-Instance GPU
Confidential Computing
PCIe Gen5

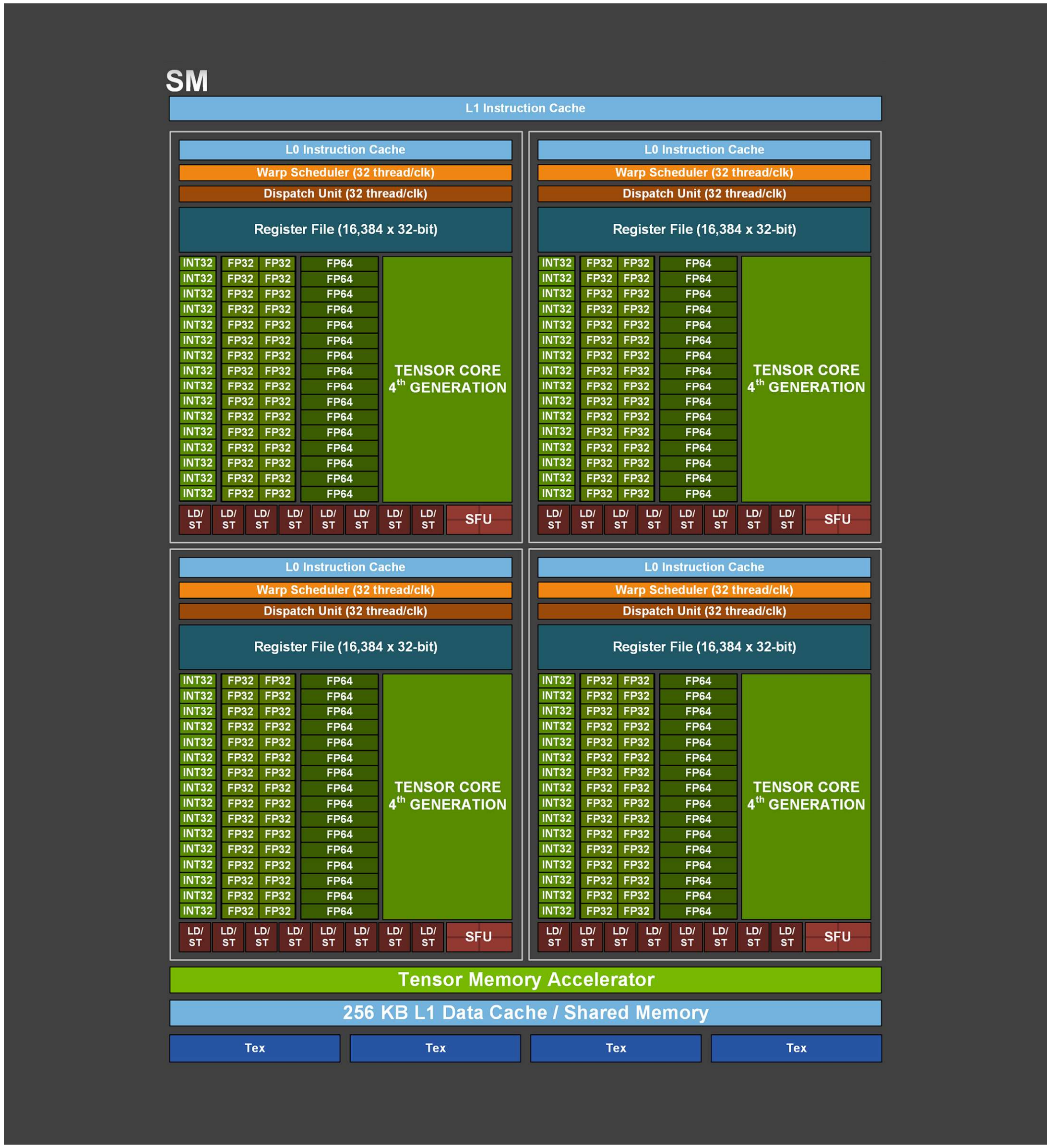
New Memory System
World's First HBM3 DRAM
Larger 50MB L2

132 SMs 2x Performance per Clock
4th Gen Tensor Core
Thread Block Clusters

4th Gen NVLink 900GB/s total BW
New SHARP support
NVLink Network

NEW HOPPER SM ARCHITECTURE

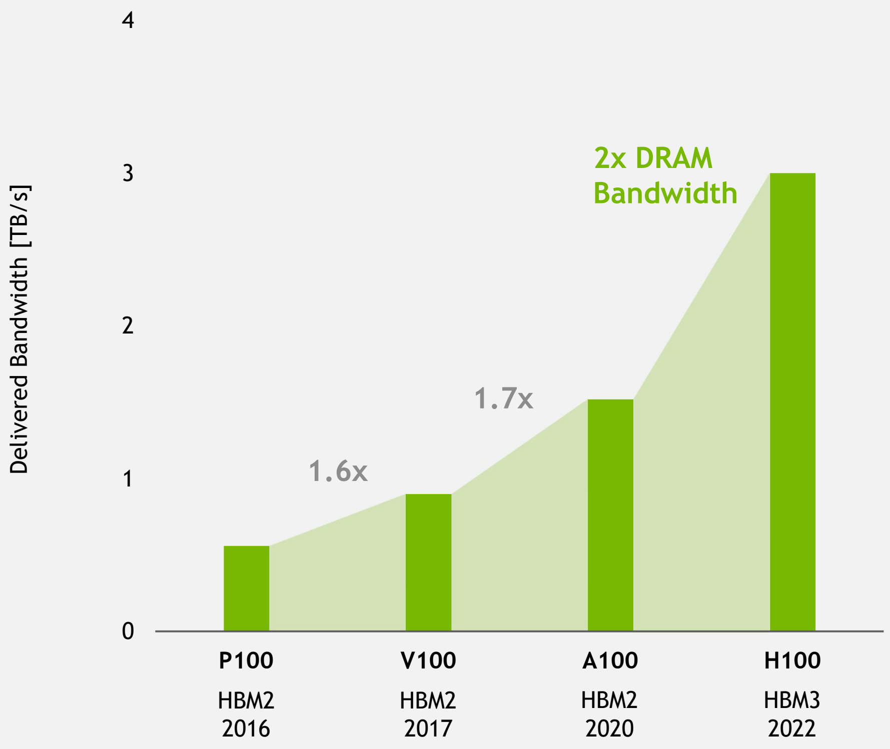
- 2x faster FP32 & FP64 FMA
- 256 KB L1$ / Shared Memory
- New 4th Gen Tensor Core
- New DPX instruction set
- New Tensor Memory Accelerator
  - Fully asynchronous data movement
- New Thread Block Clusters
  - Turn locality into efficiency


SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
INT32
FP32
FP64
TENSOR CORE
4th GENERATION
LD/ ST
SFU
Tensor Memory Accelerator
256 KB L1 Data Cache / Shared Memory
Tex
Tex
Tex
Tex

WORLD'S FIRST HBM3 MEMORY ARCHITECTURE

Greatest Generational Leap in Memory Bandwidth 3 TB/s

- 5 HBM sites with 80 GB capacity
- Dramatic improvement in HBM frequency
- New DRAM controller with 2x independent channels maintains same high efficiency



Memory data rates not finalized and subject to change in the final product.

HOPPER H100 MULTI-INSTANCED GPUS

Faster and More Secure

Higher perf per MIG

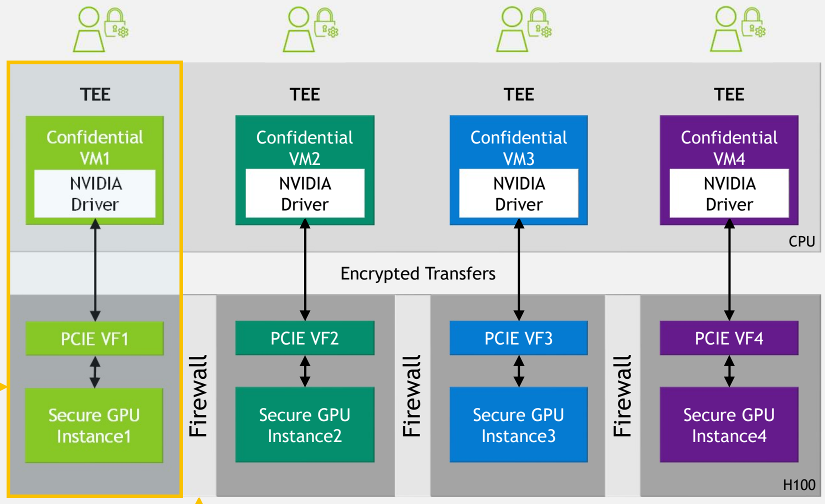
- 3X more compute capacity
- 2X more memory bandwidth

Dedicated image and video decoders per MIG

Trusted Execution Environment per MIG

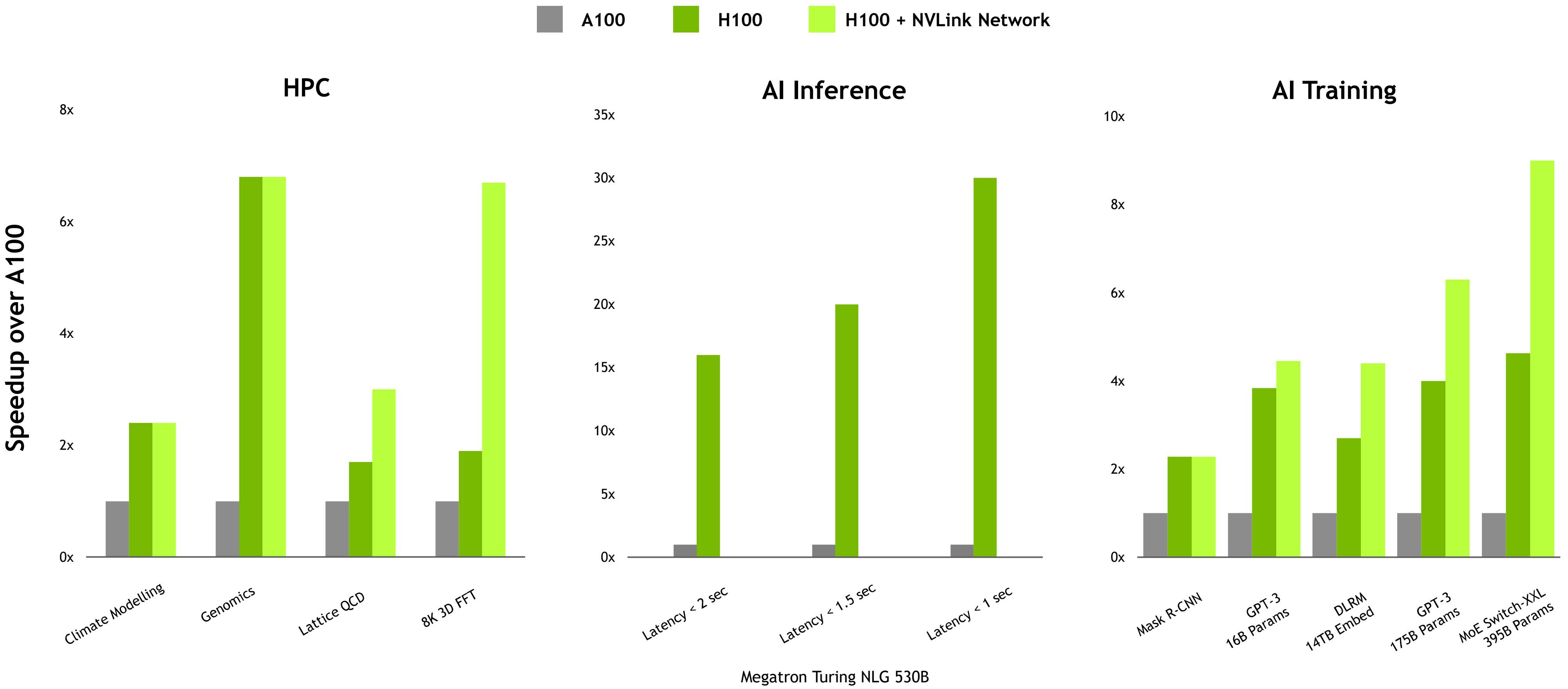
- GPU virtualization (PCIe SR-IOV)
- HW-based security for confidentiality and integrity
- HW firewalls for mem isolation between MIGs

Multi-Tenant, Single GPU Support



Secure MIG:
1 VM per GPU instance

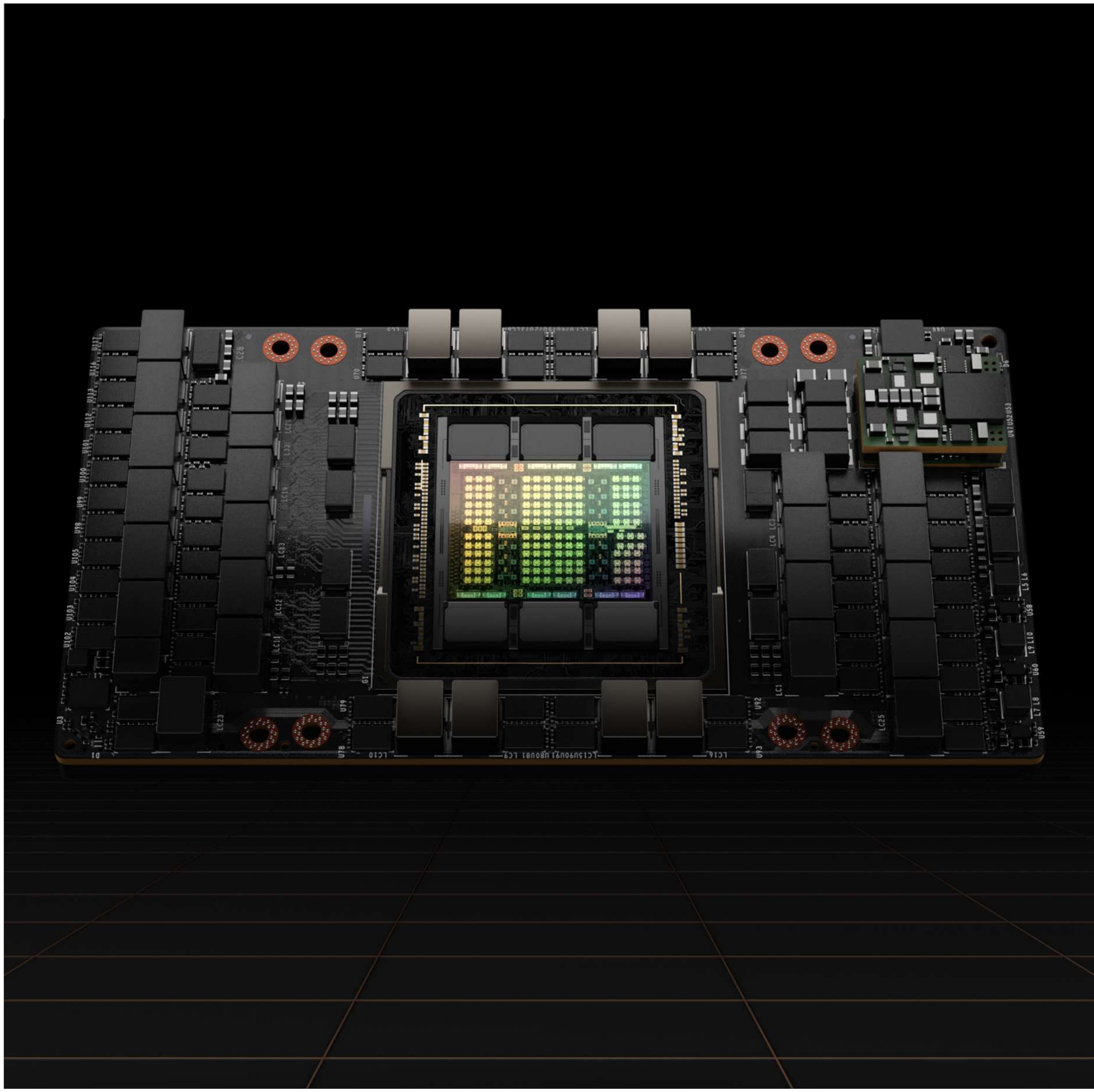
H100 ENABLES NEXT-GENERATION AI AND HPC BREAKTHROUGHS



Projected performance subject to change. A100 cluster: HDR IB network. H100 cluster: NDR IB network with NVLink Network where indicated.
# GPUs: Climate Modelling 1K, LQCD 1K, Genomics 8, 3D-FFT 256, MT-NLG 32 (batch sizes: 4 for A100, 60 for H100 at 1sec, 8 for A100 and 64 for H100 at 1.5 and 2sec), MRCNN 8 (batch 32), GPT-3 16B 512 (batch 256), DLRM 128 (batch 64K), GPT-3 175B 16K (batch 512), MoE 8K (batch 512, one expert per GPU)

AGENDA

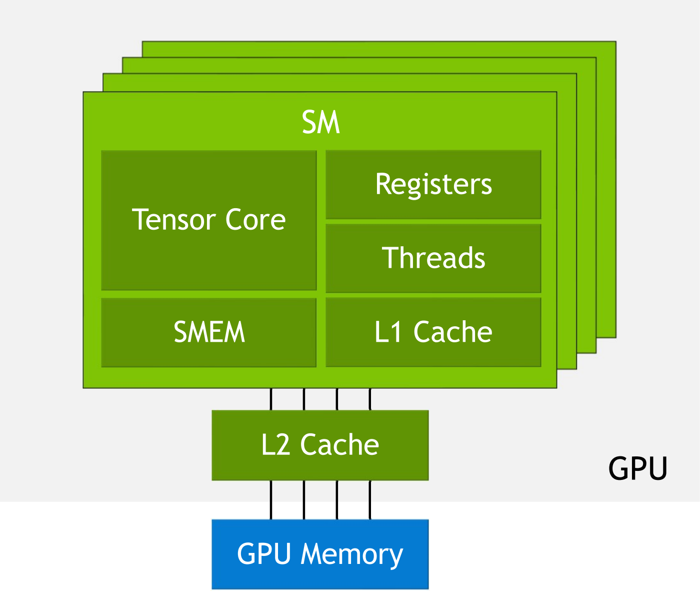
- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



KEYS TO PARALLEL PROGRAMMING PERFORMANCE

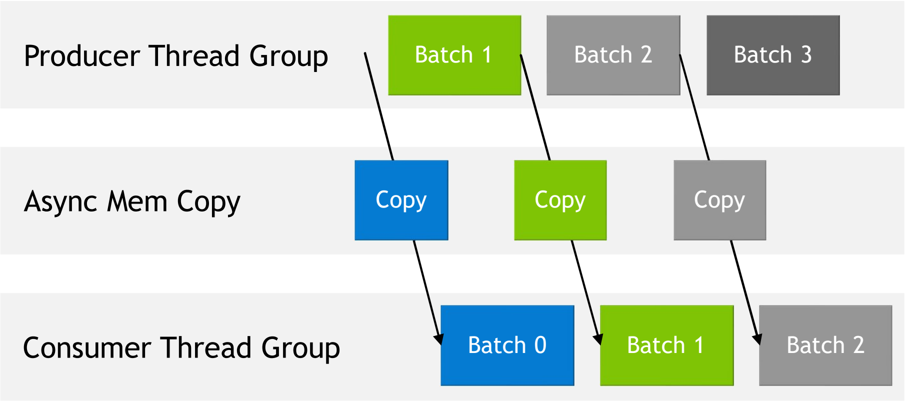
Data Locality

- Latency reduction for parallelized computation
- Higher bandwidth due to localized communication



Asynchronous Execution

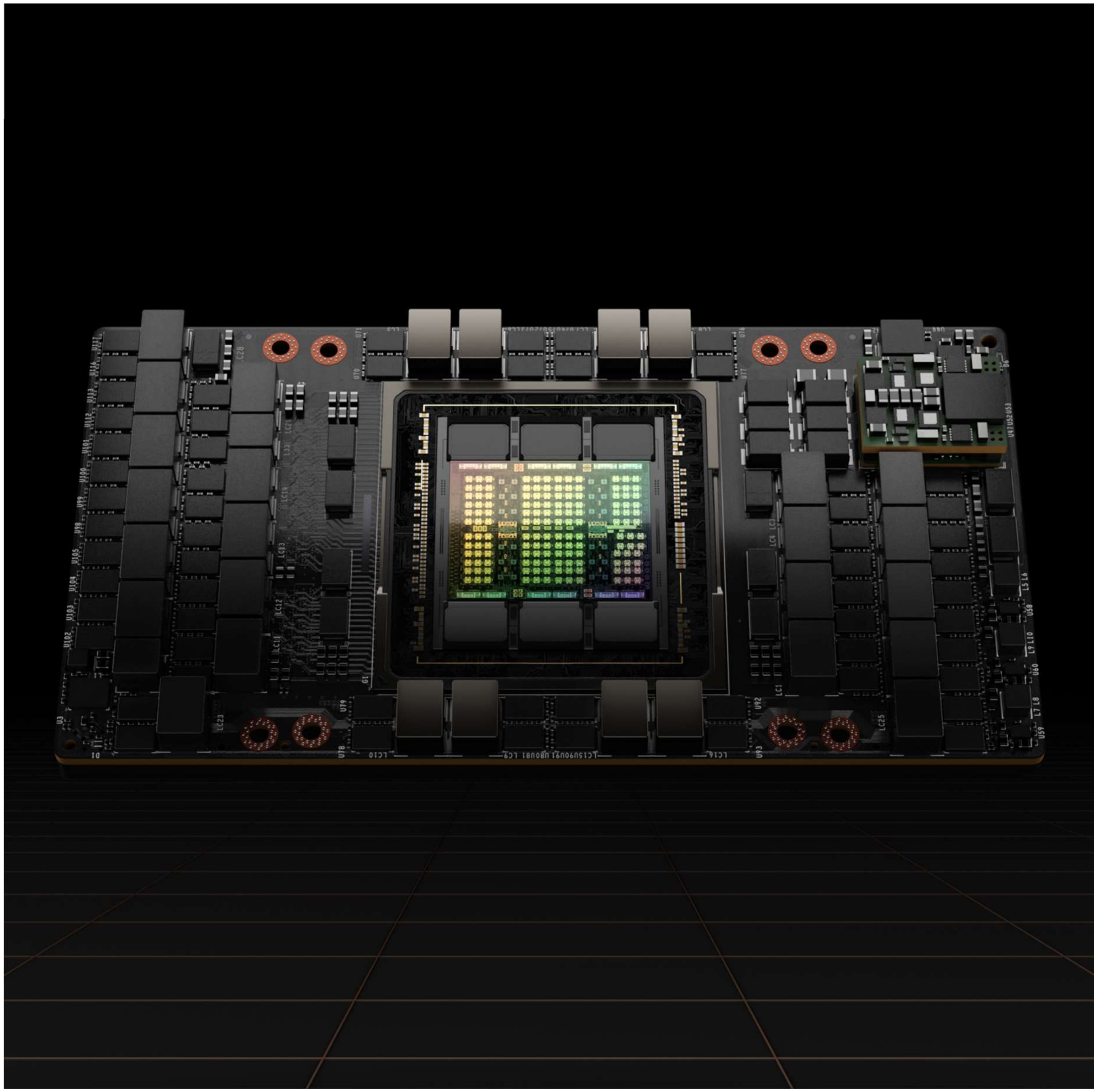
- Overlap independent work
- Keep all units fully utilized
- Concurrency with minimal synchronization delays



Overlap Memory Transfers & Processing

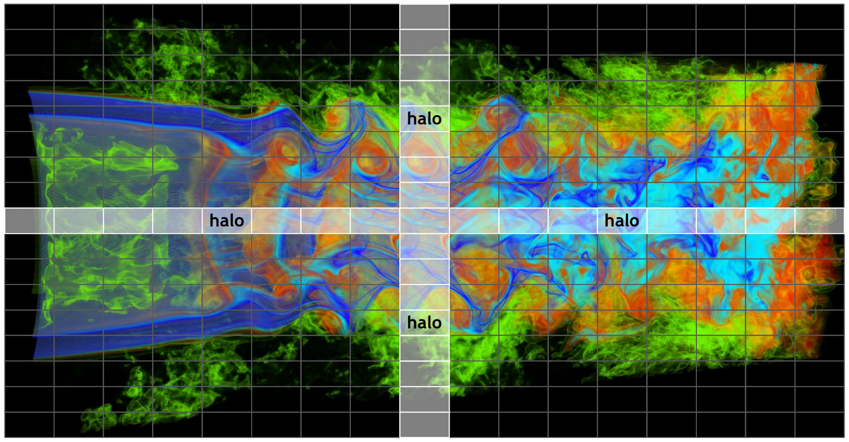
AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



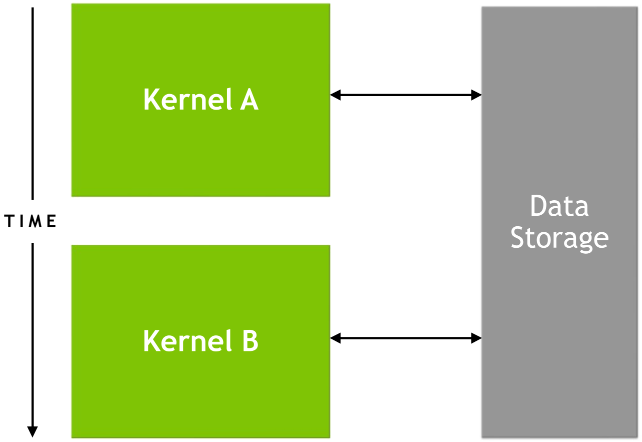
LOCALITY

Spatial



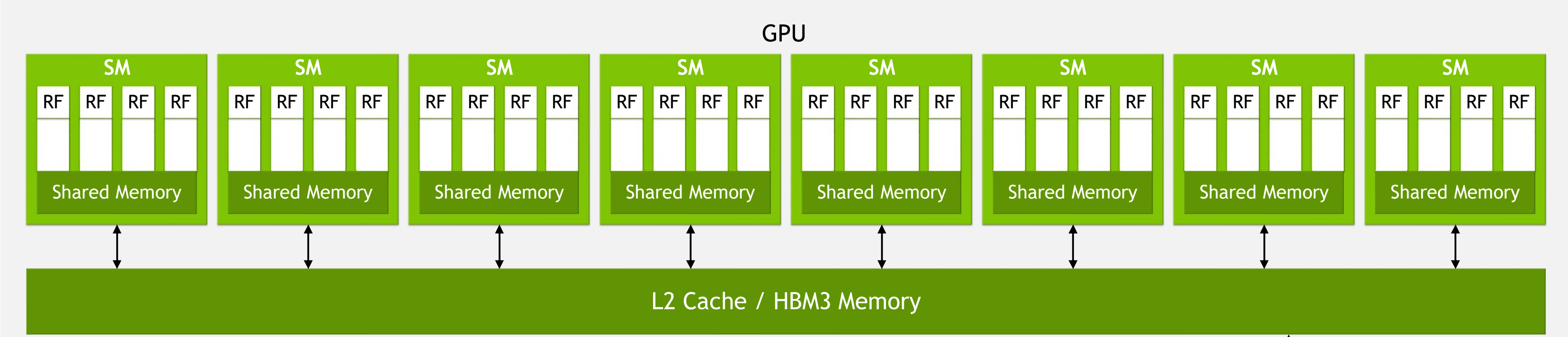
- Data & parallel execution has a spatial relationship
- Computational reuse of data, e.g.
  - Halo overlap
  - Share data in one dimension, different data in other dimension

Temporal



- Data & parallel execution has a temporal relationship
- Computation passing over data
  - One kernel processes data, then a different kernel processes the data

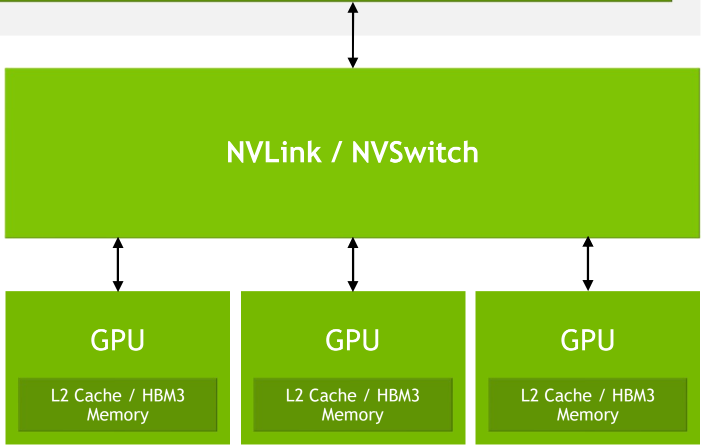
SPATIAL LOCALITY: EXISTING



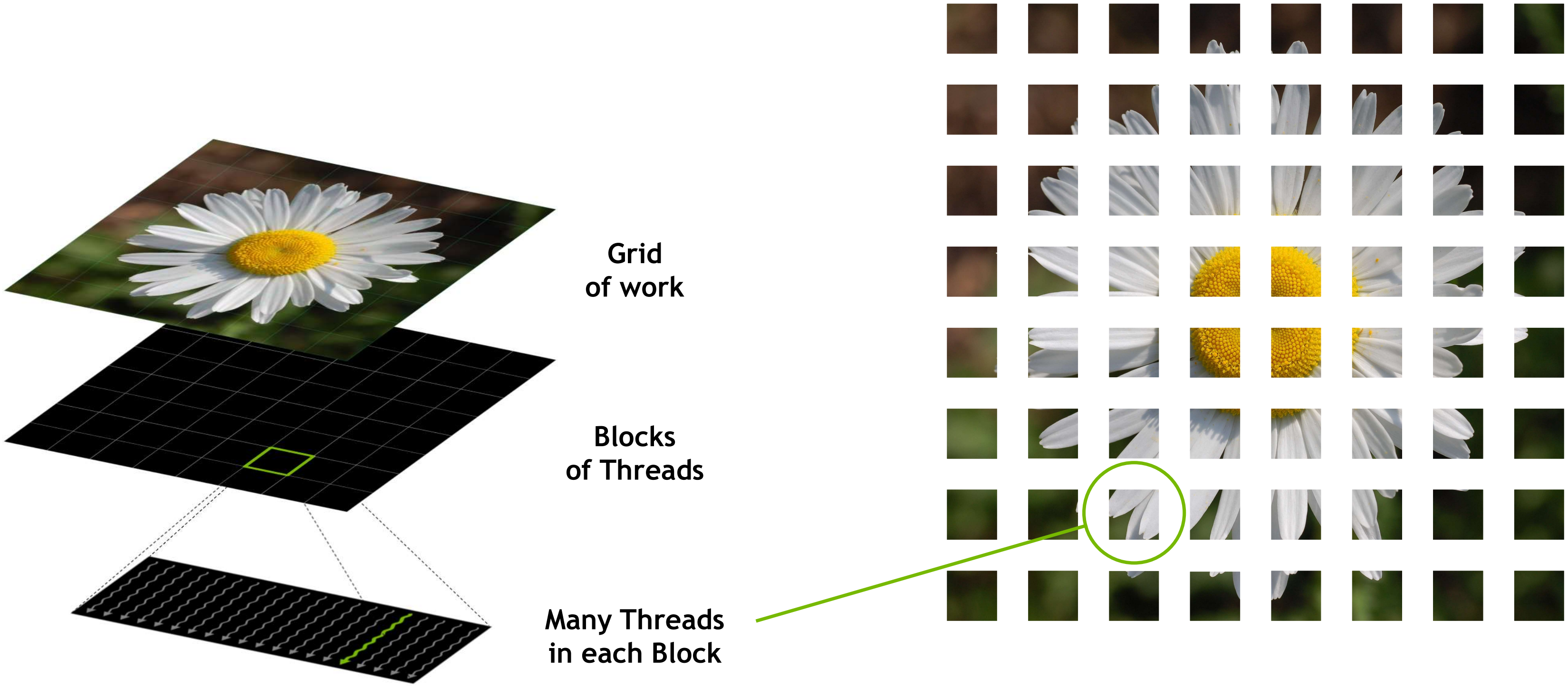
CUDA View	HW Hierarchy
Grid	GPU
Thread Block	SM
Threads	Lanes

Localized Storage

- GPU
  - 80 GB HBM3 Memory
  - 50 MB L2 Cache
- SM
  - 256KB L1 Data Cache plus Shared Memory
- Threads
  - 1KB RF per thread; 64KB per SM Partition; 256KB per SM



WORK MAPPING



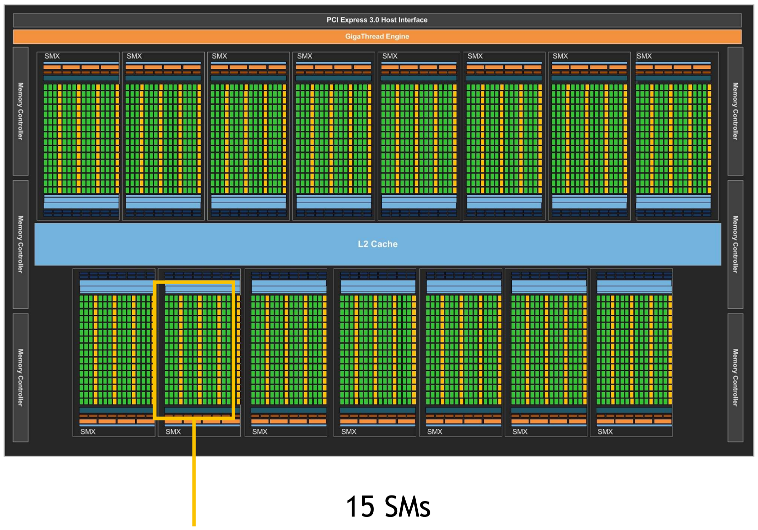
Grid
of work

Blocks
of Threads

Many Threads
in each Block

ORDERS OF MAGNITUDE GPU SCALING

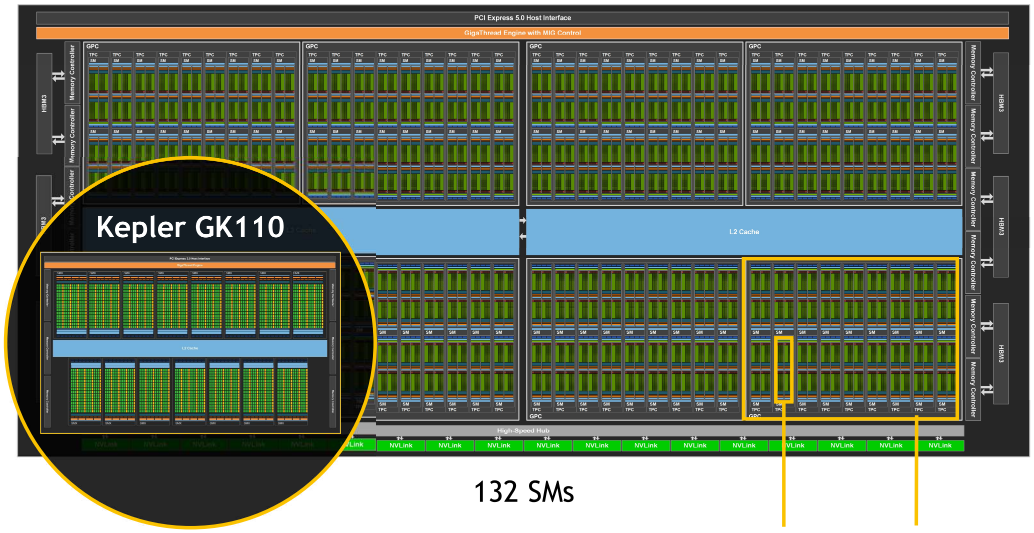
Kepler GK110 GPU, 2012



15 SMs

SM

Hopper H100 GPU, 2022

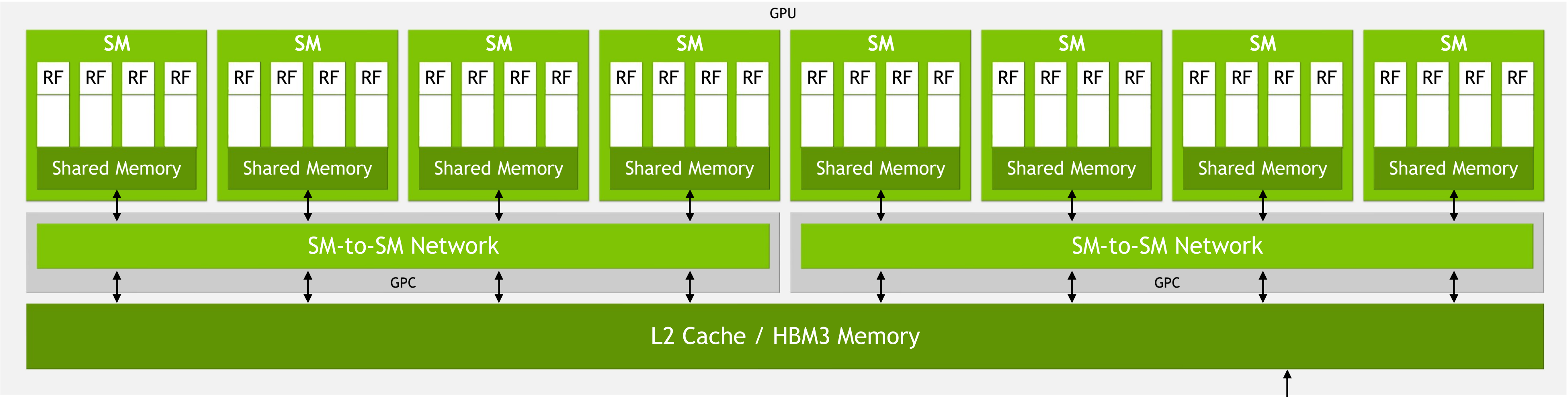


132 SMs

SM

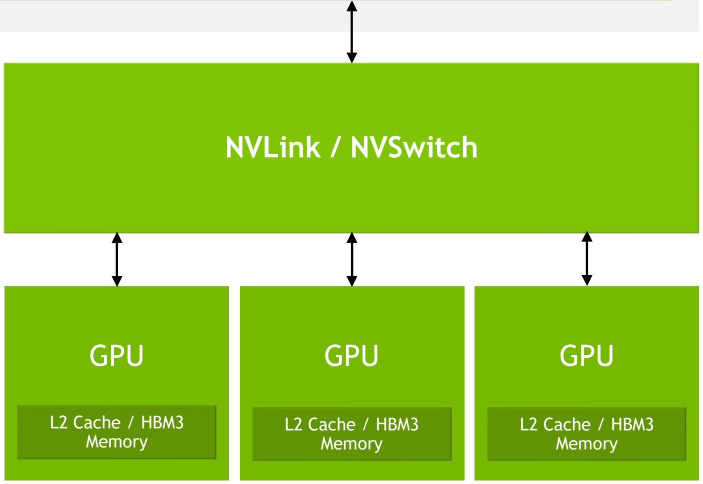
GPC

SPATIAL LOCALITY: THREAD BLOCK CLUSTERS



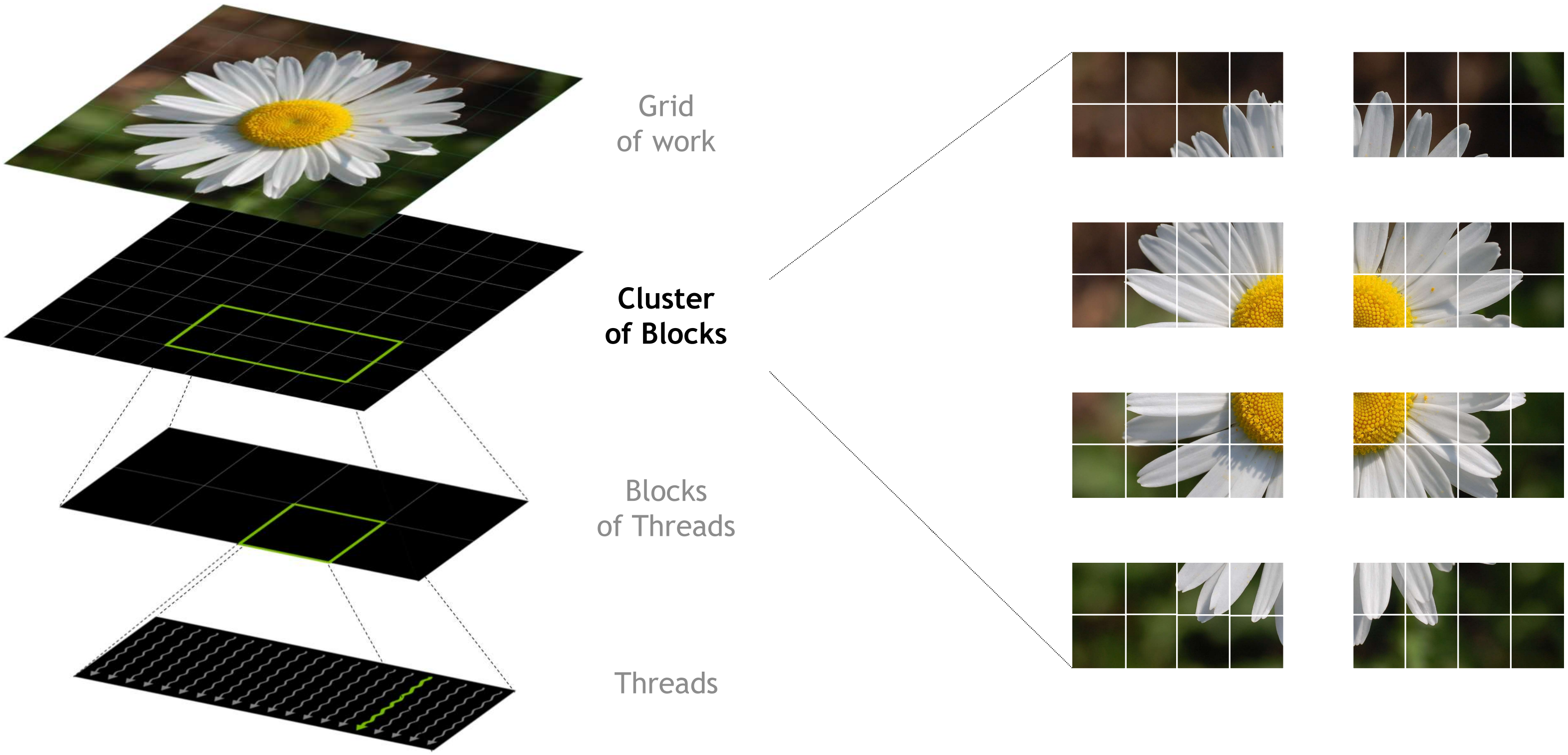
CUDA View	HW Hierarchy
Grid	GPU
Cluster	GPC
Thread Block	SM
Threads	Lanes

- Localized Storage
- GPU
  - 80 GB HBM3 Memory
  - 50 MB L2 Cache
- SM & GPC
  - 256KB L1 Data Cache plus Shared Memory per SM
- Threads
  - 1KB per thread; 64KB per SM Partition; 256KB per SM



THREAD BLOCK CLUSTER

A Collective of Blocks, Co-scheduled on Adjacent Multiprocessors



Grid of work

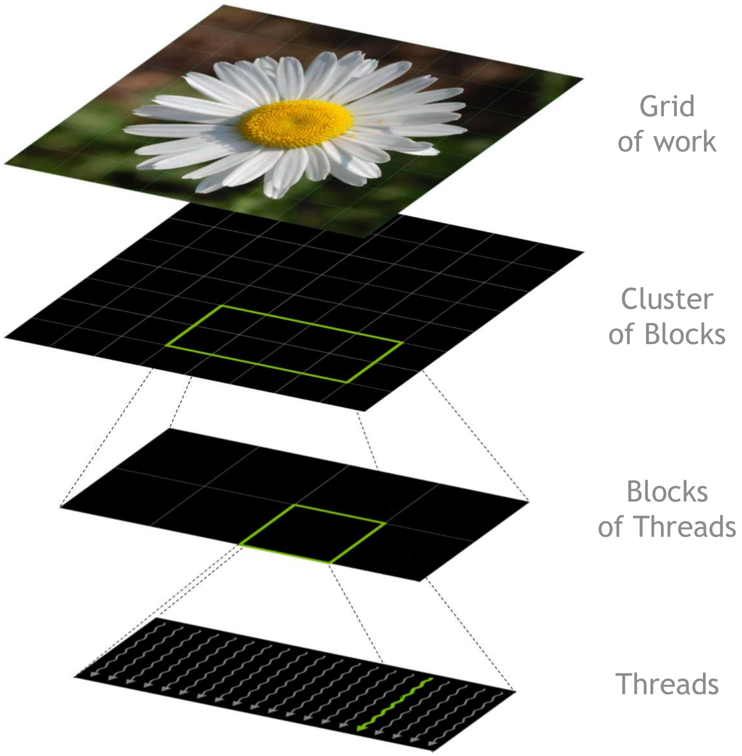
Cluster of Blocks

Blocks of Threads

Threads

THREAD BLOCK CLUSTER

Building Hierarchy into a Program



A cluster is a collective of up to 16 blocks

Guaranteed to be on different SMs

Guaranteed to be running at the same time

1D, 2D or 3D, just like blocks

Annotate a kernel with its required cluster size

New cluster dimension annotation for __global__ functions:

`__cluster_dims__(x, [y, [z]])`

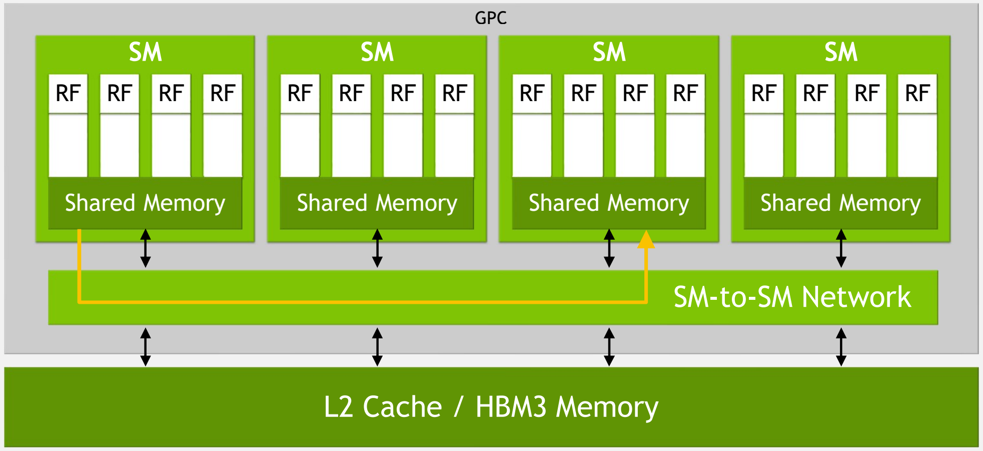
```
__cluster_dims__(4, 2, 1)           // 8-block cluster of size
4x2x1
__global__ void helloCluster()
{
    cooperative_groups::cluster_group cluster = this_cluster();
    cluster.sync();

    printf(“Hello from cluster elem %d\n”, cluster.cluster_rank());
}
```

Plus: New extensible launch API allows configuration at launch time

DIRECT SM-TO-SM COMMUNICATIONS WITHIN A CLUSTER

- Dedicated SM to SM network for direct low latency access w/out needing to go through L2
- Threads can reference another Thread Block’s Shared Mem directly
  - Distributed Shared Memory (DSMEM) Programming Model, laid out as a Partitioned Global Address Space
  - Loads, stores, atomics, reductions, asynchronous DMA ops, Arrive barrier ops
- Accelerated Synchronization and Data Exchange
  - Blocks in a cluster can synchronize together via barriers in DSMEM
  - Asynchronous DMA operations



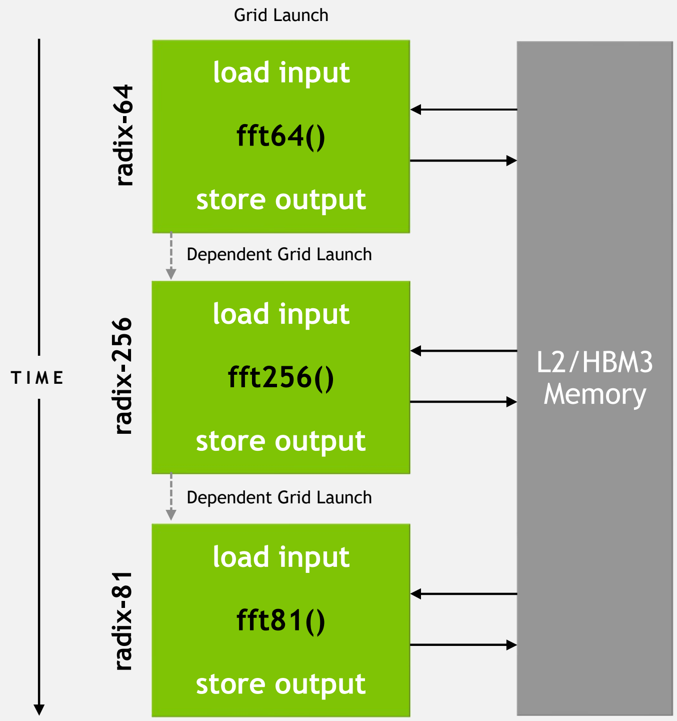
TEMPORAL LOCALITY: EXISTING

- Data moved into Local HBM3 memory
- Multiple dependent kernels operate on that data

Limitation/Challenges

- Dependent kernels must be separate launches
- Any data locally stored in SM must be flushed
  - to L2/HBM3 memory between kernels

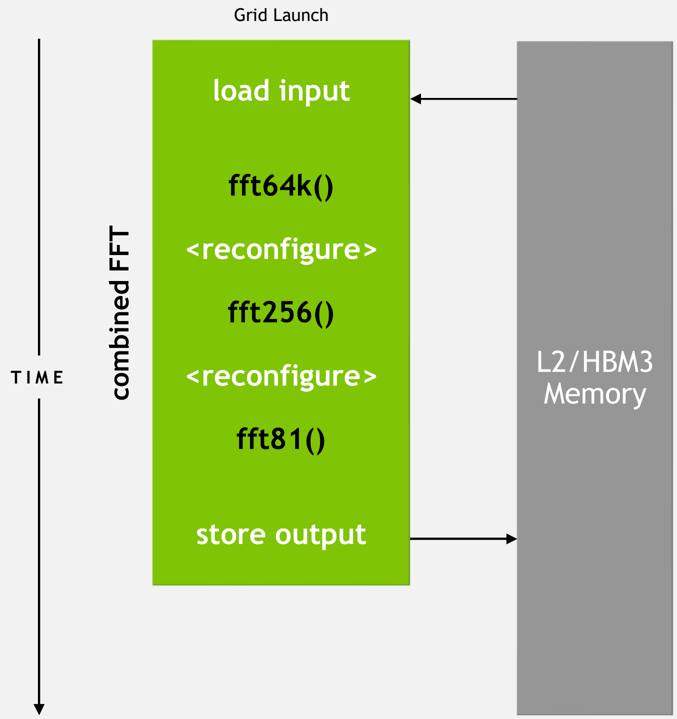
FFT Workflow



TEMPORAL LOCALITY: THREAD BLOCK RECONFIGURATION

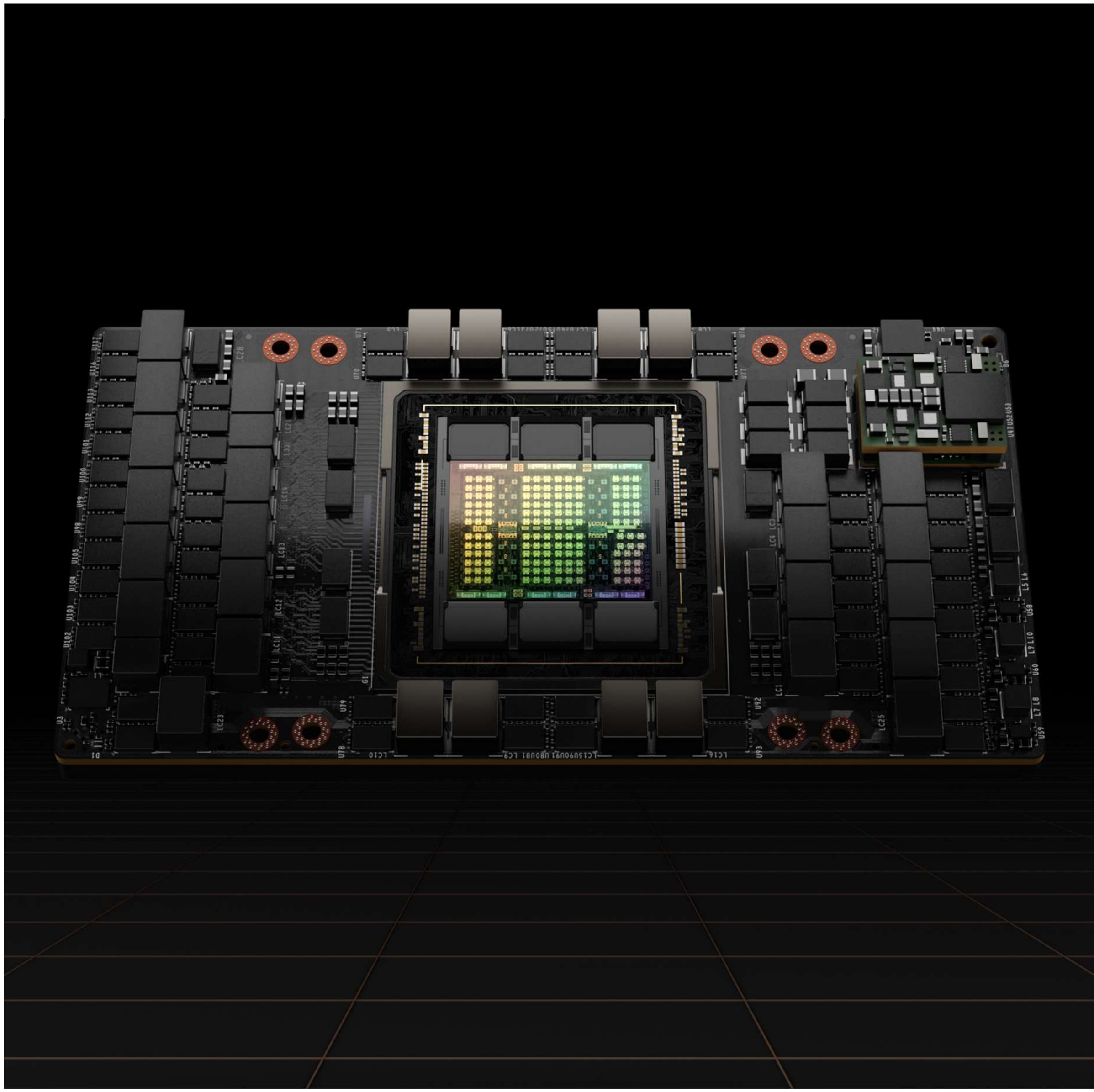
- Data moved into Local SMEM/DSMEM
- Multiple dependent kernels operate on that data
- Each kernel able to change thread count and RF allocation per thread in most efficient work to thread mapping
- Data stays resident in SMEM/DSMEM between kernels

FFT Workflow



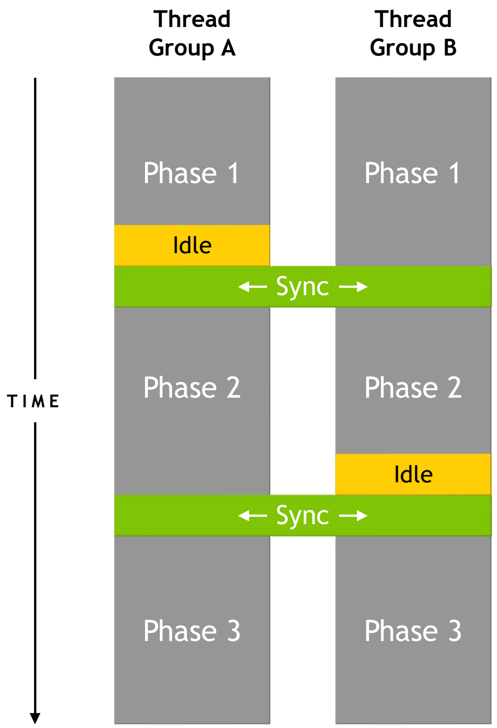
AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



SYNCHRONOUS MACHINE

Cooperative Execution



Thread Group A

Thread Group B

TIME

Phase 1

Phase 1

Idle

← Sync →

Phase 2

Phase 2

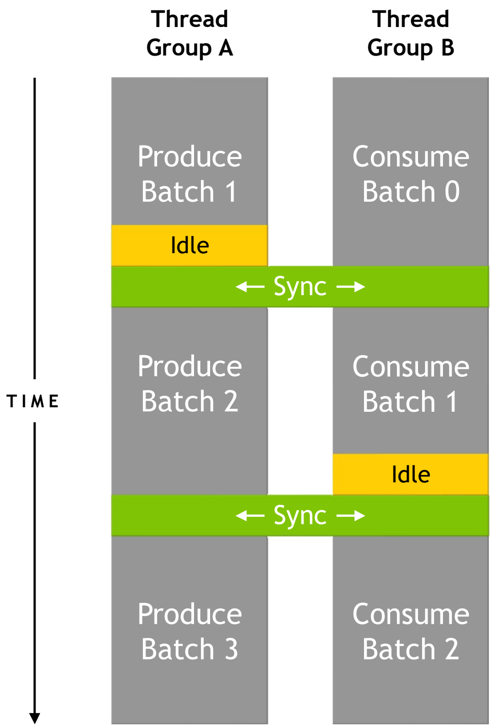
Idle

← Sync →

Phase 3

Phase 3

Producer/Consumer Pipeline



Thread Group A

Thread Group B

TIME

Produce Batch 1

Consume Batch 0

Idle

← Sync →

Produce Batch 2

Consume Batch 1

Idle

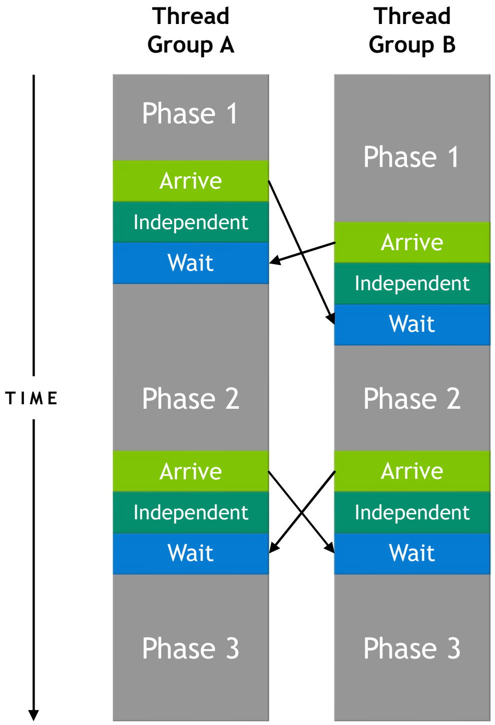
← Sync →

Produce Batch 3

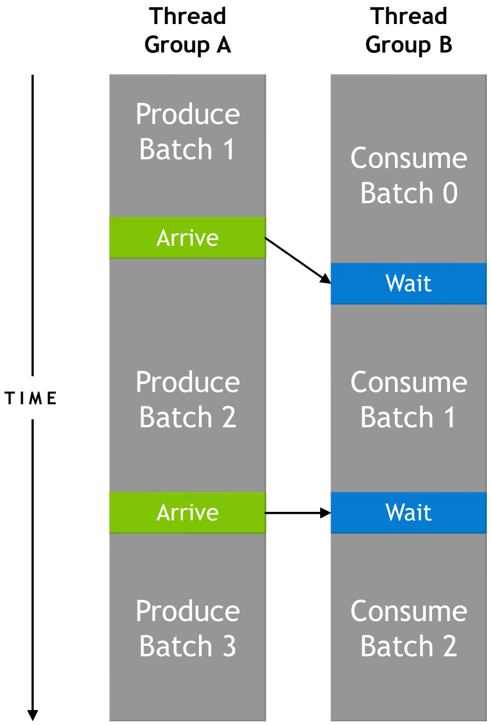
Consume Batch 2

ASYNCHRONOUS MACHINE

Cooperative Execution



Producer/Consumer Pipeline



ASYNCHRONOUS BARRIER

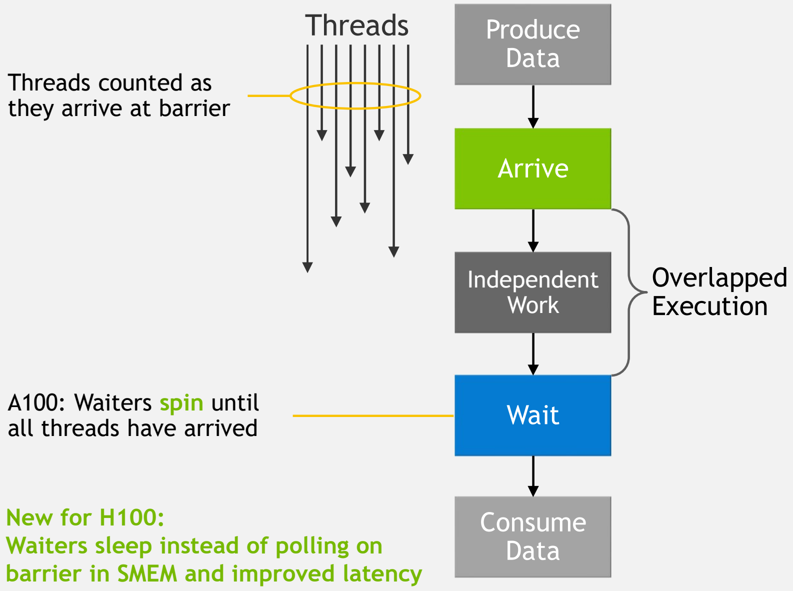
Permits Overlapped Execution of Independent Work

- Produce data > Barrier > Consume data
- Barrier split into 2 steps
  - Arrive = Thread done producing data
  - Wait = Thread ready to start consuming data
- Arrive is non-blocking

Use cases

- Synchronizing with other threads in Block
- Synchronizing with other thread in Cluster

Asynchronous Barrier (from A100)


Threads
Threads counted as they arrive at barrier
Produce Data
Arrive
Independent Work
Overlapped Execution
A100: Waiters spin until all threads have arrived
Wait
Consume Data

New for H100:
Waiters sleep instead of polling on barrier in SMEM and improved latency

ASYNCHRONOUS TRANSACTION BARRIER

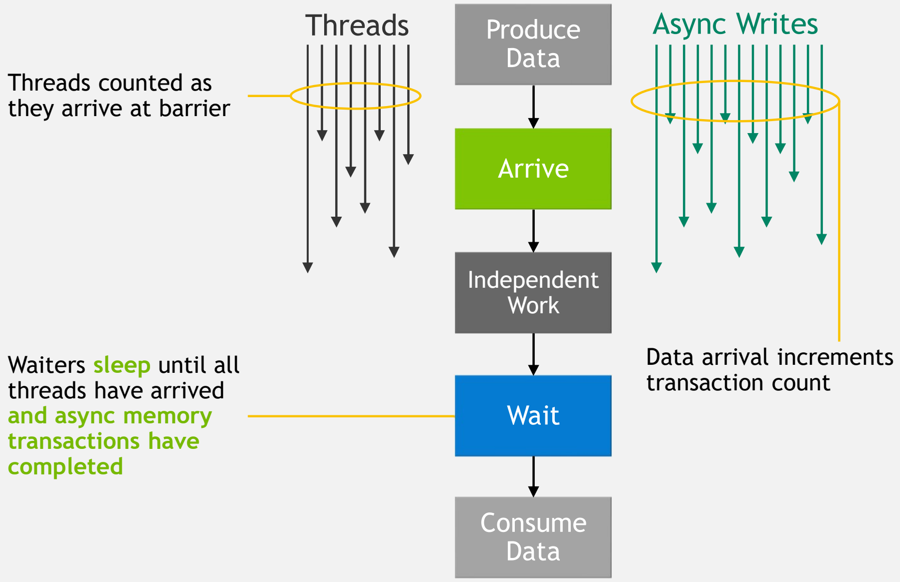
New Form of Barrier with “Data Arrival Tracking”

- Barrier counts threads and async memory transactions
- Store passes data + transaction_count
- Drop-in enhancement to existing cuda::barrier

Use cases

- Cluster Block to Block communication with barrier
- Async Mem_copy with barrier

Async Transaction Barrier (New on H100)

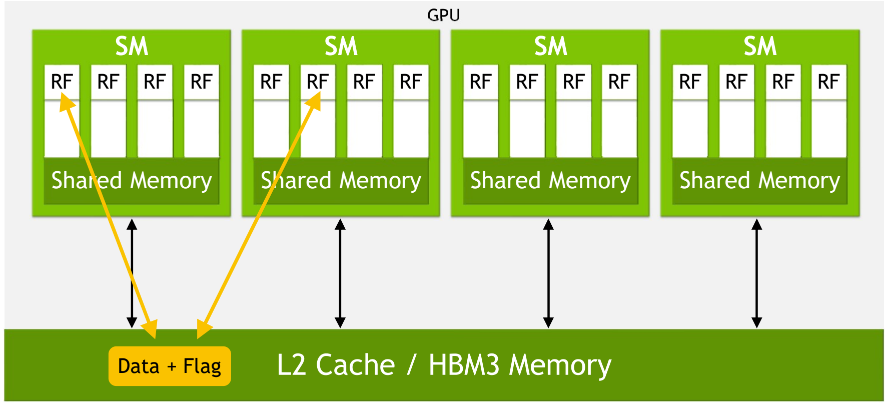


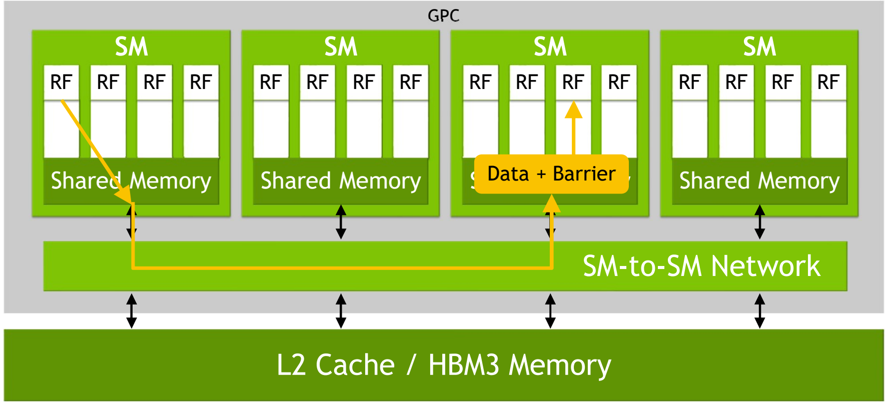
Threads counted as they arrive at barrier

Data arrival increments transaction count

Waiters sleep until all threads have arrived and async memory transactions have completed

BLOCK TO BLOCK DATA EXCHANGE





Existing: Data Exchange via Global Memory

Exchange requires 3-4 round-trips to global mem

- Write data*
- Memory barrier
- Write flag
- Poll flag (request & response)
- Read data (request & response)

New: Asynchronous Store within Cluster

Exchange requires only a one-way trip to DSMEM

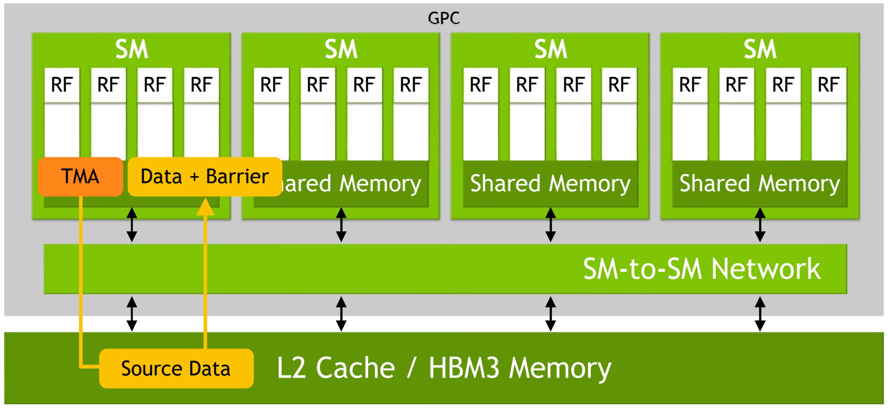
Minimum latency data exchange

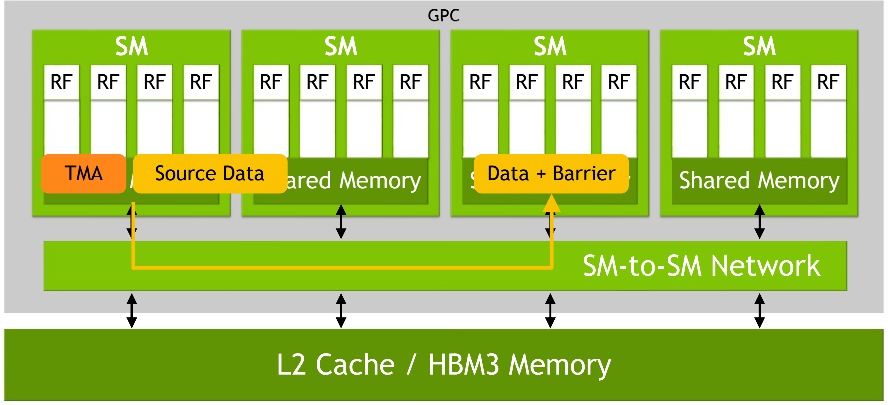
7x latency reduction

- Write data* and update barrier

*Both stores and reduction atomics supported

ASYNC MEM COPY USING TMA





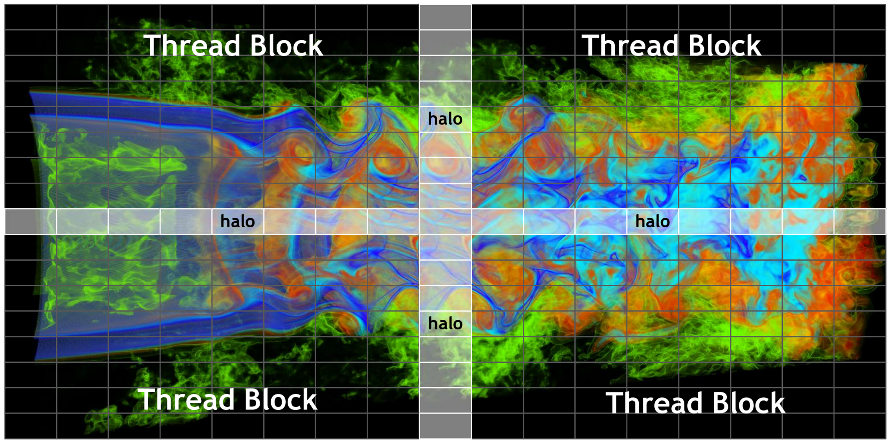
HW-accelerated mem_copies

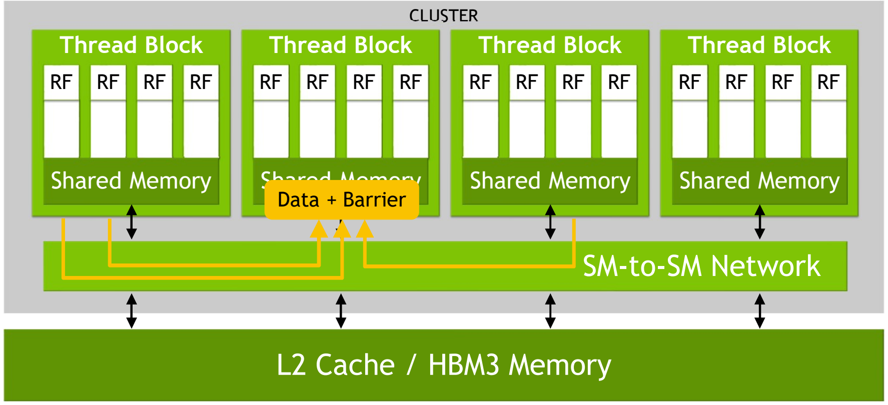
- Global <=> Shared Mem
- Shared Mem <=> Shared Mem for Clusters
- Address generation for 1D to 5D Tensors

Fully asynchronous with respect to threads

- No addr gen or data movement overhead
- Synchronize with transaction barrier
- Simplified programming model

EXAMPLE HALO DATA EXCHANGE



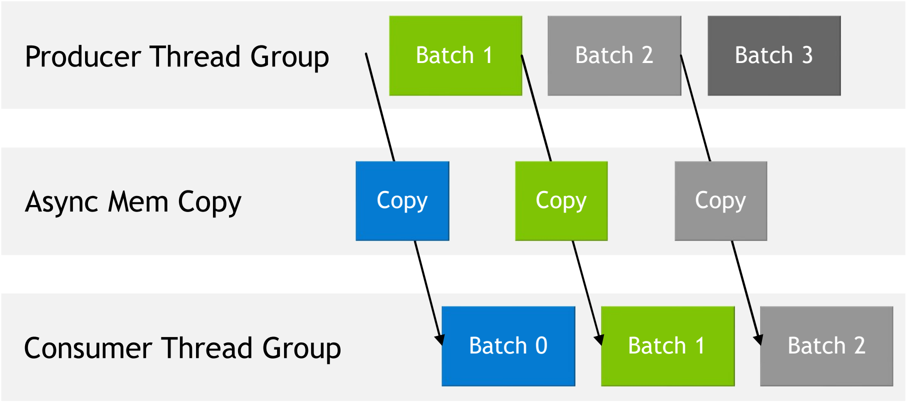


Efficient asynchronous data exchange with minimal latency

A FULLY ASYNCHRONOUS GPU ARCHITECTURE

Hopper Enables End-to-End Fully Asynchronous Pipelines

- Async Transaction Barriers — Atomic data movement with synchronization
- More efficient Waiting on Barriers
- Async Mem_copy via TMA

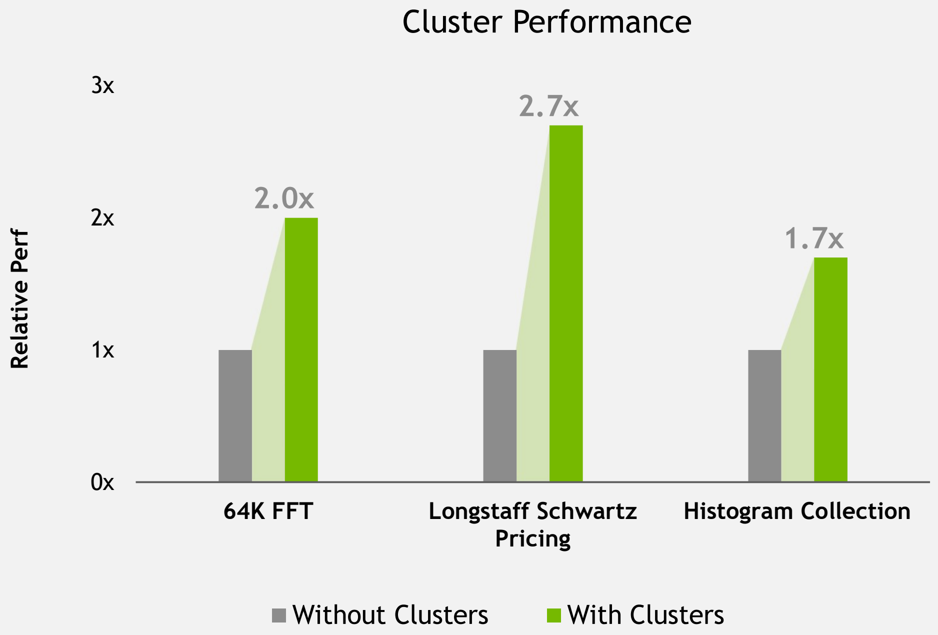


CLUSTERS AND ASYNC EXECUTION

Programmatically Exploiting the Hierarchy of the GPU

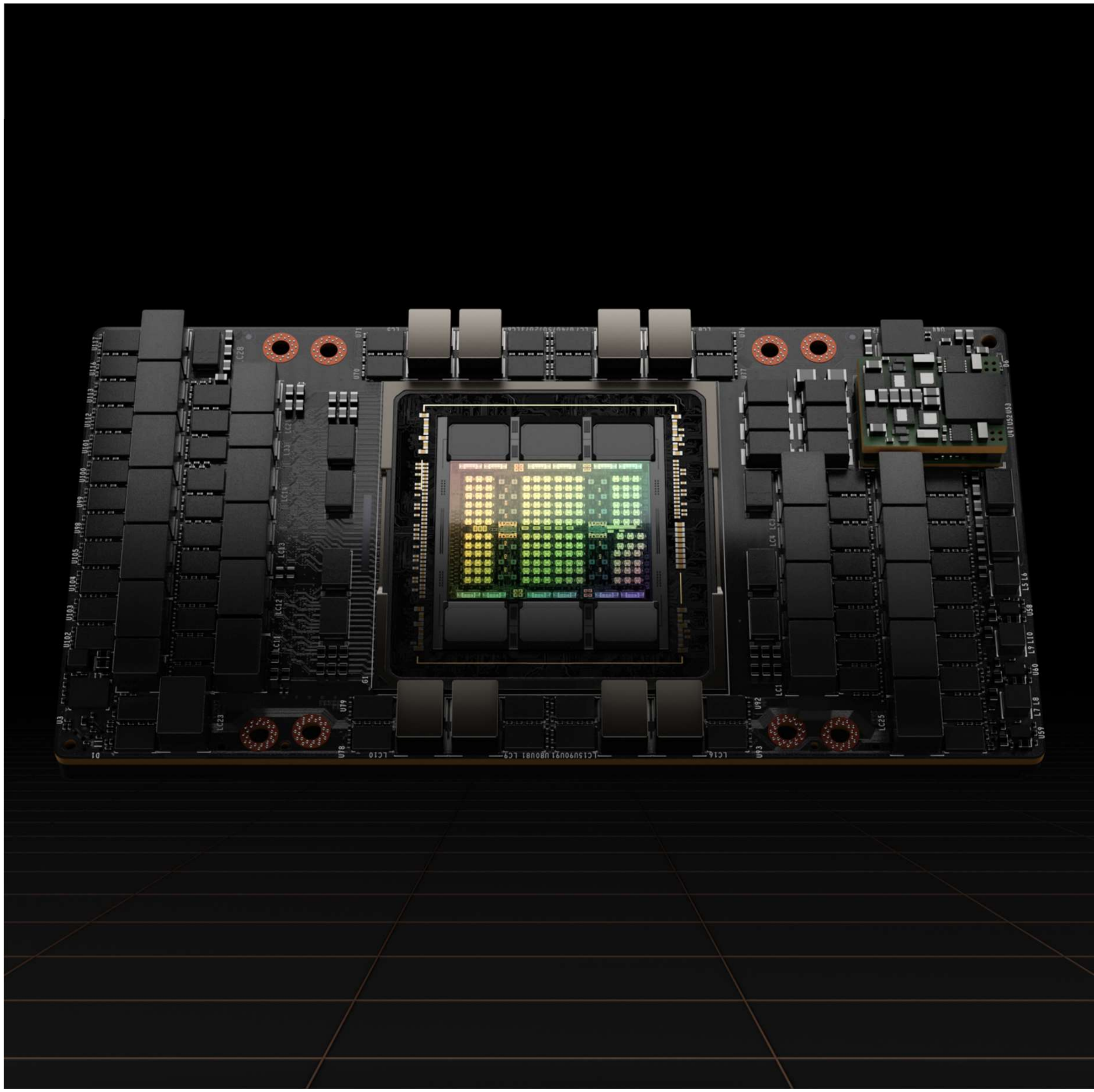
- Thread Block Clusters
- Fast synchronization
- Inter-Block Shared memory access (DSMEM)
- Minimum latency data exchange with transaction barrier
- TMA async memory copy

Cooperative execution with more threads & larger shared mem, combined with asynchronous execution & data movement yields higher perf



AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up

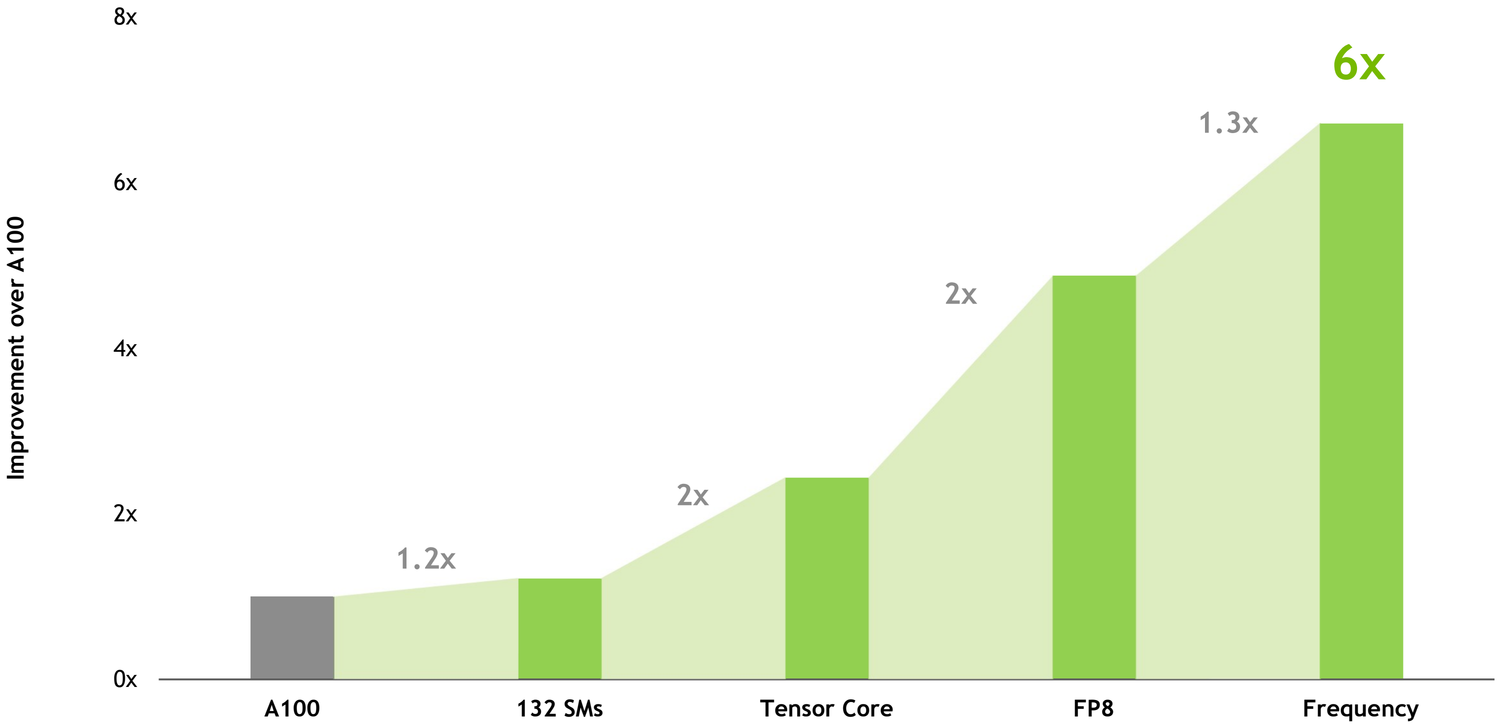


HOPPER 4TH GEN TENSOR CORE

- 2x faster clock-for-clock
- Supports wide range of storage and math formats
- New FP8 format support
- More efficient data management saves up to 30% operand delivery power
- Accelerates sparse tensor arithmetic

Format	A100 SM MACs/clock dense | sparse	H100 SM MACs/clock dense | sparse	Speedup
FP64	64 | ---	0128 | ------	2x
TF32	512 | 1024	1024 | 2048	2x
FP16	1024 | 2048	2048 | 4096	2x
BF16	1024 | 2048	2048 | 4096	2x
INT8	2048 | 4096	4096 | 8192	2x
FP8	-	4096 | 8192	New!

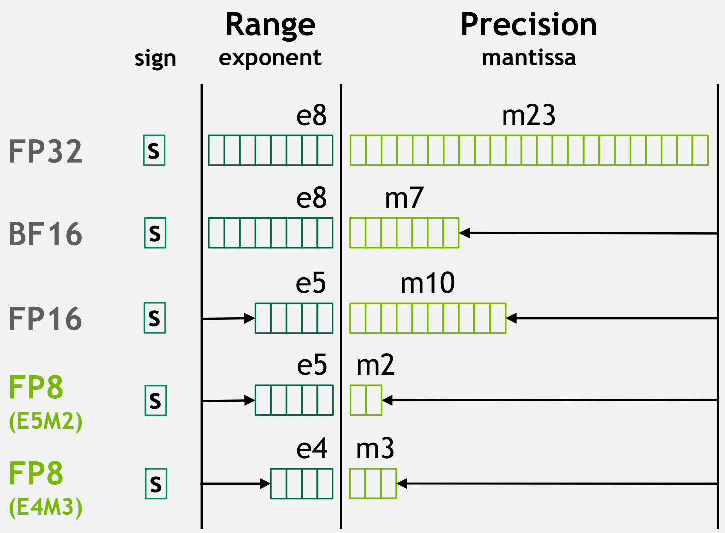
H100 COMPUTE IMPROVEMENTS BREAKDOWN



6x throughput for the world's most compute-hungry workloads

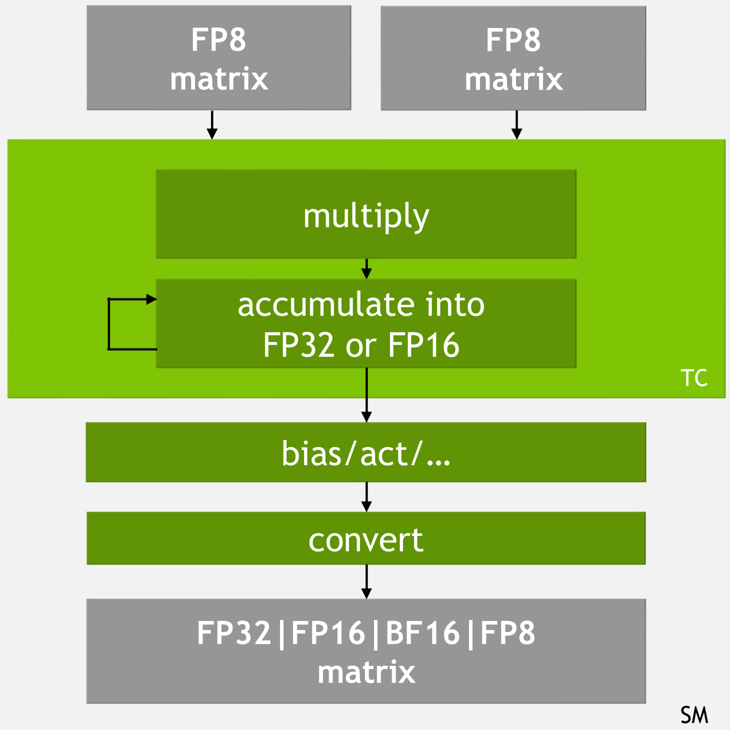
FP8 TENSOR CORE

Allocate 1 bit to either range or precision



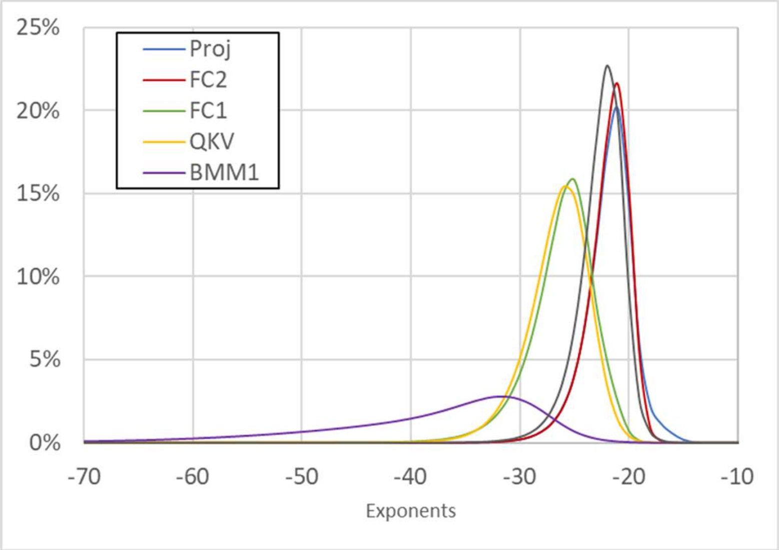
2x throughput & half footprint of FP16/BF16

Support for multiple accumulator and output types

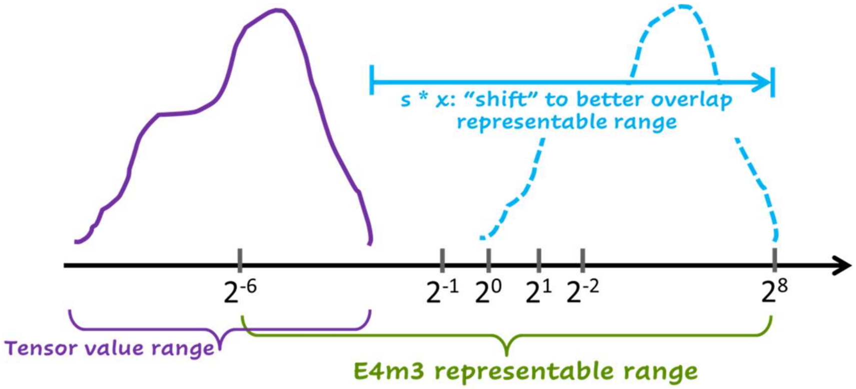


FP8 NUMERICS

- E4M3: needed for forward pass/inference (2-bit mantissa insufficient for some nets)
- E5M2: needed for some gradient tensors in some networks (E4 dynamic range not wide enough)
  - E.g. BMM1 in Transformer Attention



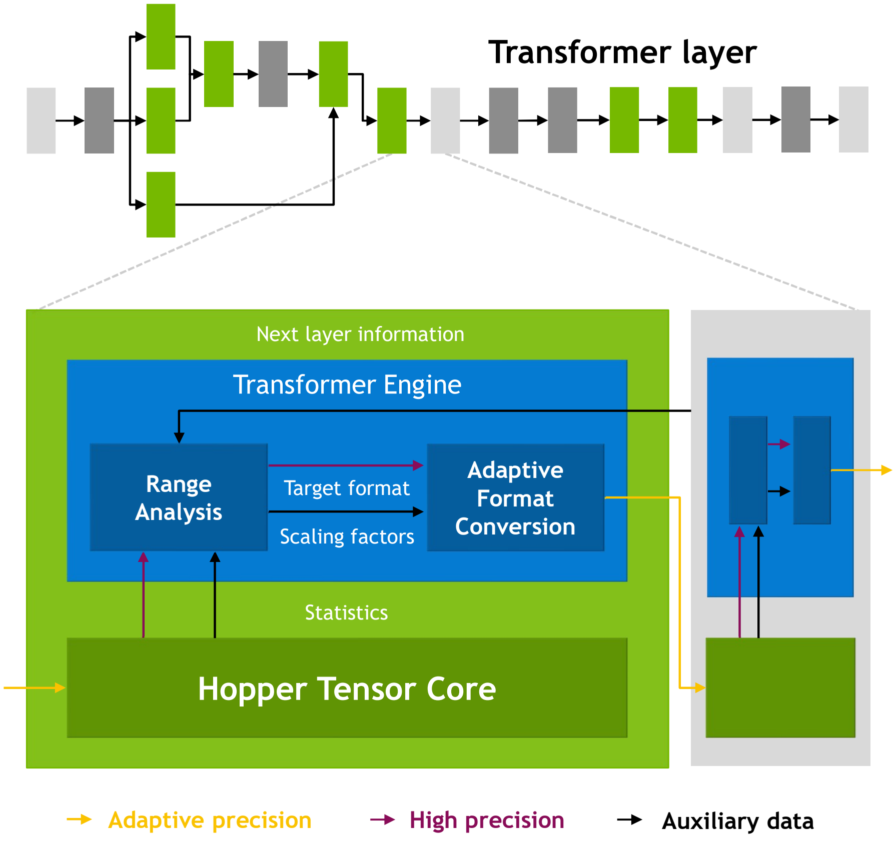
- Tensor values are computed in higher precision, converted to FP8
- Scale (i.e. “shift”) tensor values prior to FP8 conversion:



- “Unscale” after linear math (matrix multiply), prior to other math or conversions

FP8 TRANSFORMER ENGINE

- Optimal Transformer acceleration with Hopper Tensor Core
- Transparent to DL frameworks
- User can enable/disable
- Selectively applies new FP8 format for highest throughput
- Monitors tensor statistics and dynamically adjusts range to maintain accuracy

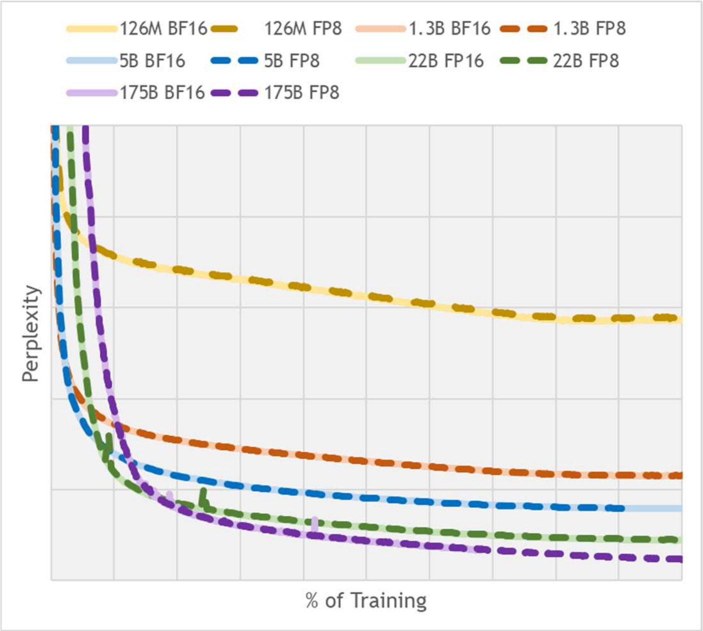


TRANSFORMER MODELS TRAINED WITH FP8

Matches 16-bit training accuracy/perplexity and downstream task performance
FP8 inference after training requires no quantization or fine-tuning

Architecture	Network	Dataset	Metric	16-bits	FP8
Transformer	**Vaswani Base**	WMT	BLEU	26.87	26.76
	Vaswani Large	WMT	BLEU	28.43	28.35
Transformer	**XL Base**	WikiText	PPL*	22.71	22.76
	XL Large	WikiText	PPL*	17.90	17.85[1]
BERT	**BERT Base**	Wikipedia	Loss*	1.352	1.357[1]
	BERT Large	Wikipedia	Loss*	1.163	1.167

GPT-3 Language Models



[1] Gradients are in E5M2, otherwise all linear inputs are E4M3

*Lower is better

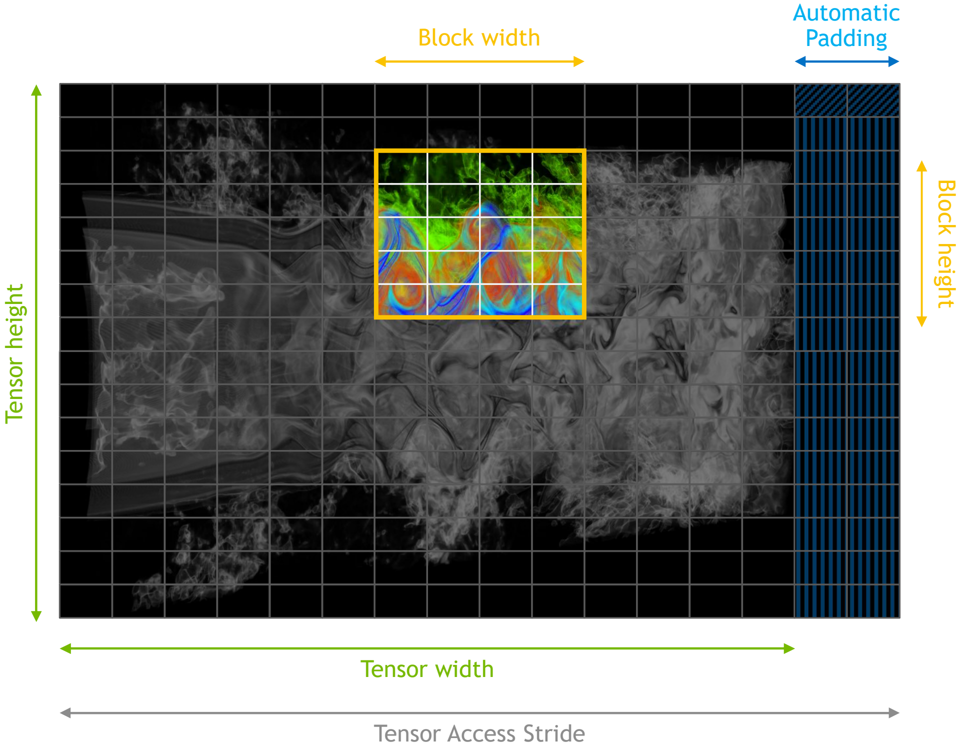
All models trained on A100 using "emulated" FP8 input/output (pre-silicon/pre-SW methodology)

TMA: EFFICIENT COPY OF DL TENSOR MEMORY

Multi-Dimensional Tensor Copying

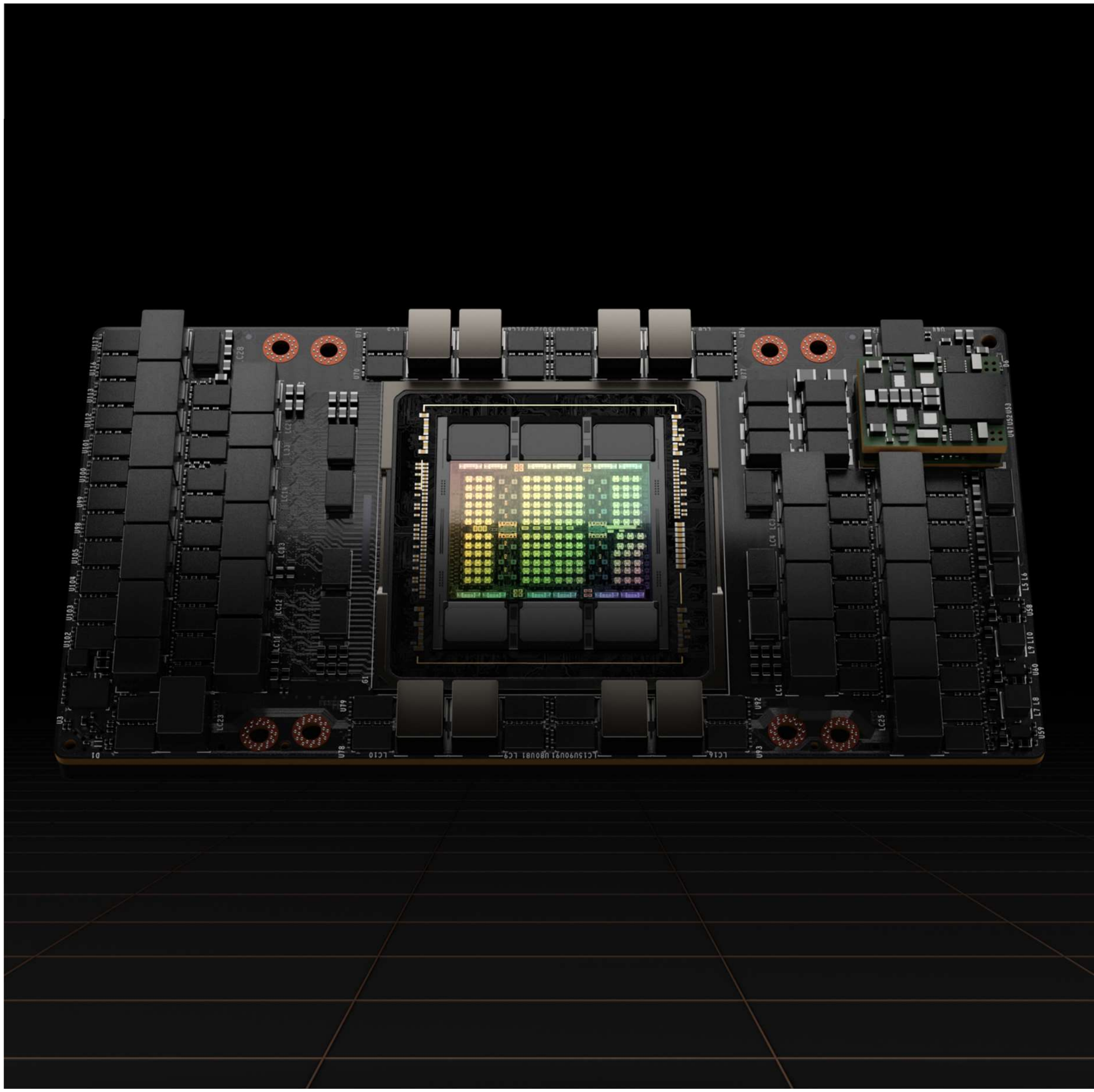
- Automatic stride & address generation up to tensors of rank 5
- Boundary padding for out-of-bounds accesses
- Fire-and-forget from a single thread – everything handled by TMA
- No iteration or bounds-checking code required

The TMA can copy sub-regions of a multi-dimensional tensor



AGENDA

- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



DGX H100 SUPERPOD: AI EXASCALE

- 32 DGX H100 nodes
- 256 H100 Tensor Core GPUs
- 164 NVLink4 NVSwitch chips
- 1 ExaFLOP peak AI compute
- 70.4 TB/s bisection bandwidth
- Network optimized for AI and HPC
- New NVLink Network interconnect
- NDR 400 Gb/s InfiniBand

Check out the NVSwitch/SuperPOD Hot Chip Talk for More Details!

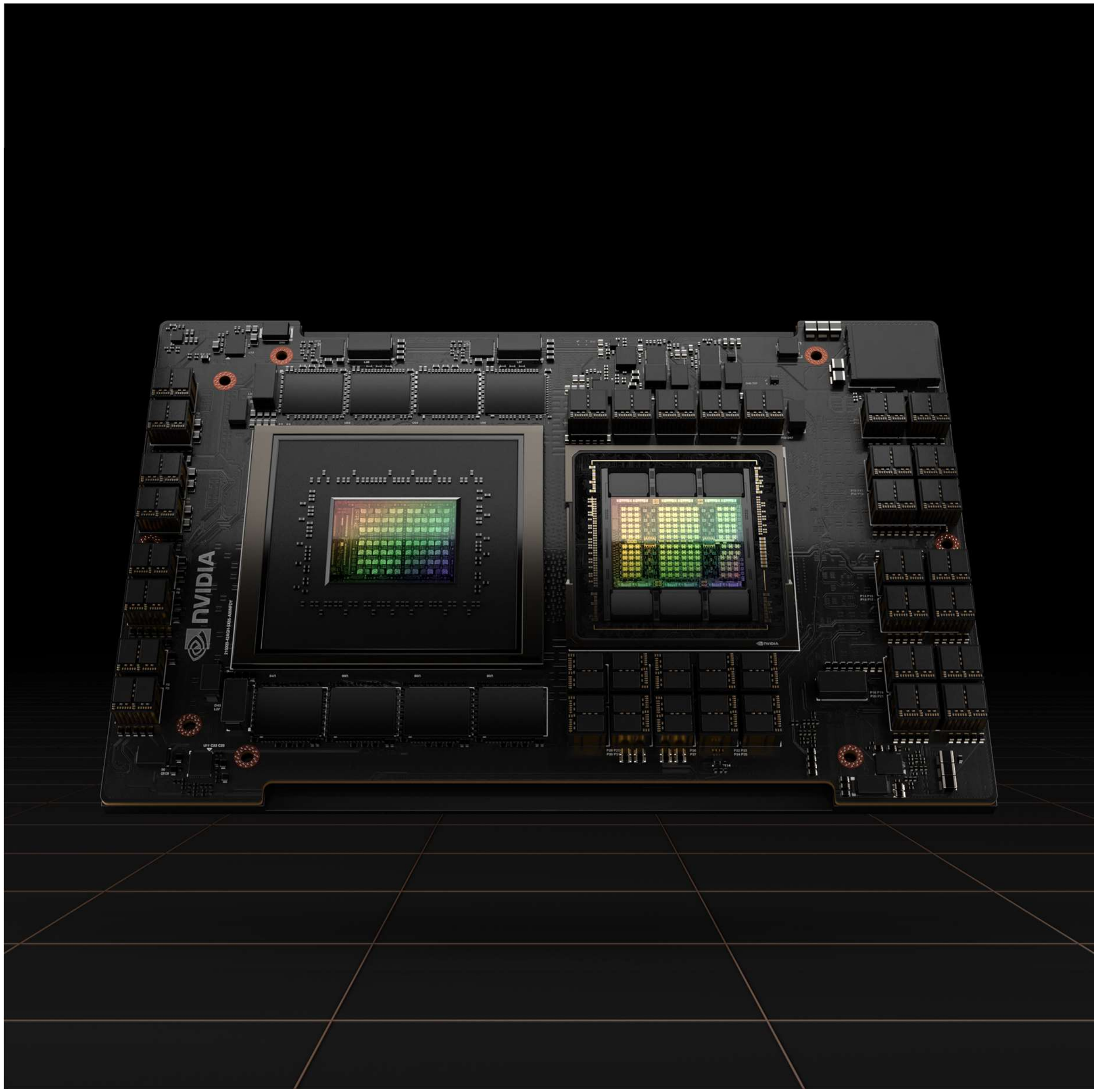
Peak compute throughput numbers assume sparse FP8

NVIDIA GRACE HOPPER

Grace CPU + Hopper GPU

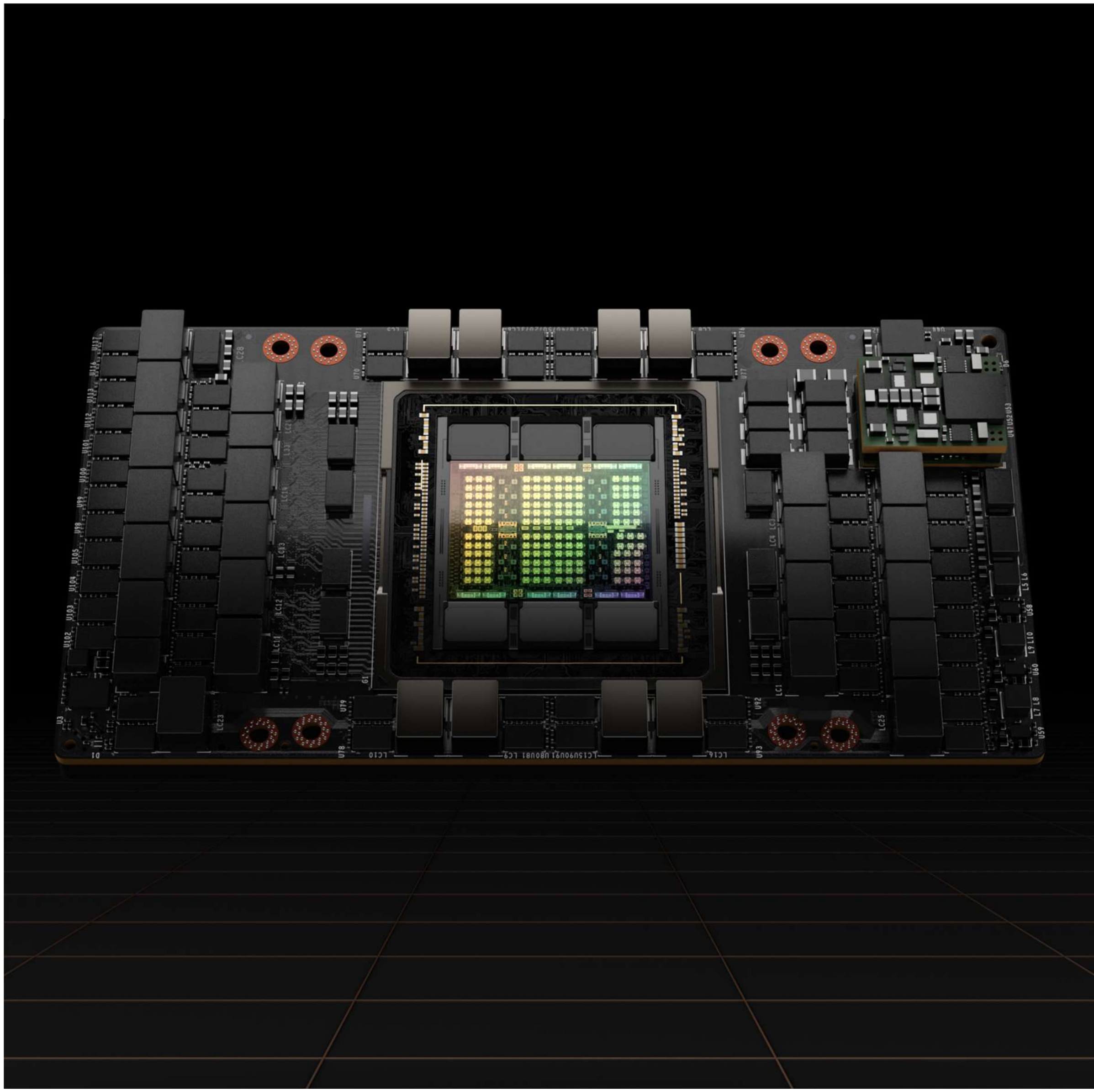
- Up to 512GB LPDDR5
  - 6x more than GPU HBM
- 900 GB/s CPU-GPU BW
  - 7x PCIe Gen5 bandwidth
  - Hardware coherent

Check out the Grace CPU Hot Chip Talk for More Details!



AGENDA

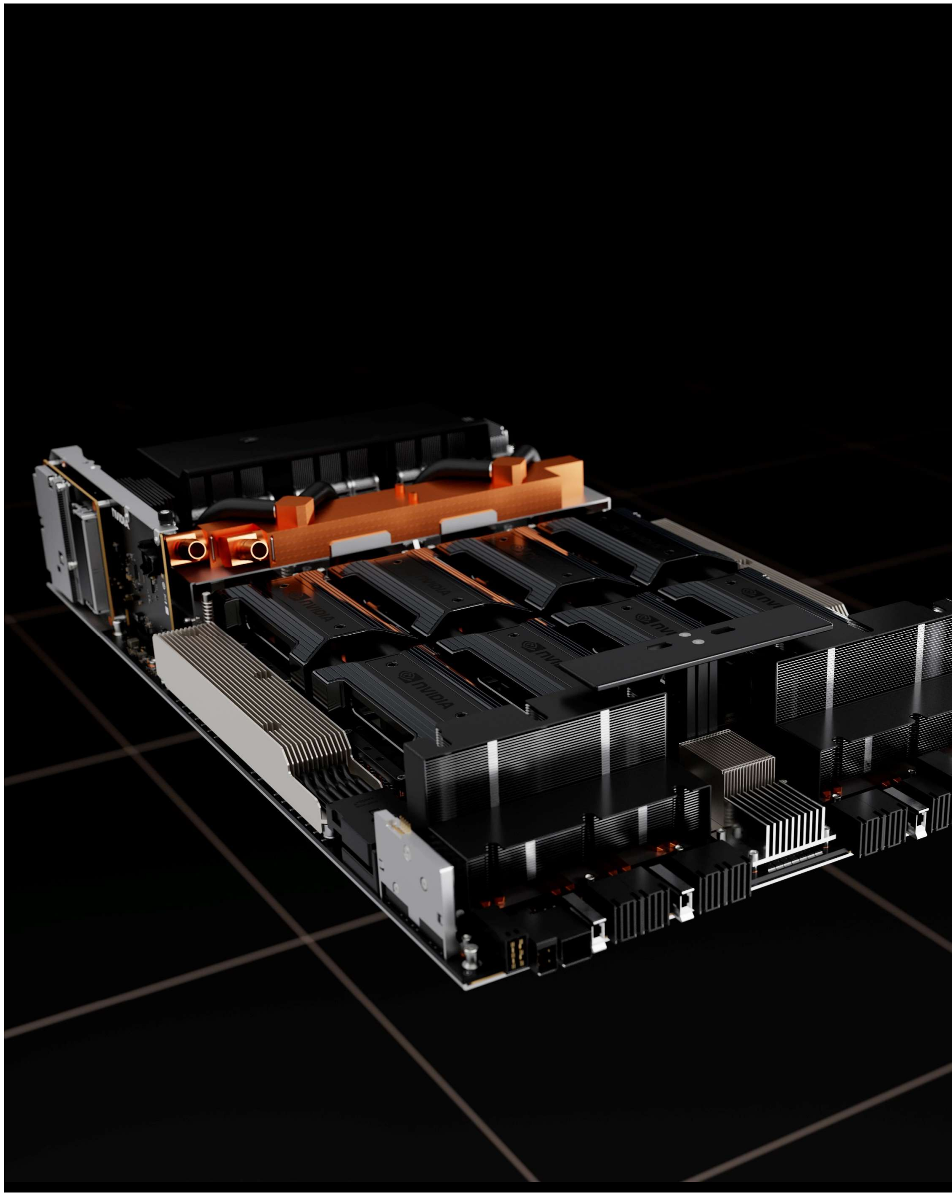
- H100 GPU Overview
- Accelerating Principles for Performance
  - Data Locality & Cooperative Execution
  - Asynchronous Execution & Data Transfer
- Accelerating Deep Learning
- Preview: Scaling Up and Out
- Wrap Up



HOPPER DELIVERS A GENERATIONAL LEAP IN PERFORMANCE, EFFICIENCY, AND SECURITY

H100 Whitepaper
www.nvidia.com/hopper-architecture-whitepaper

THANKS TO THE MANY NVIDIA ENGINEERS WHO DESIGNED AND BUILT THE H100 GPU AND THOSE WHO CONTRIBUTED TO THIS PRESENTATION

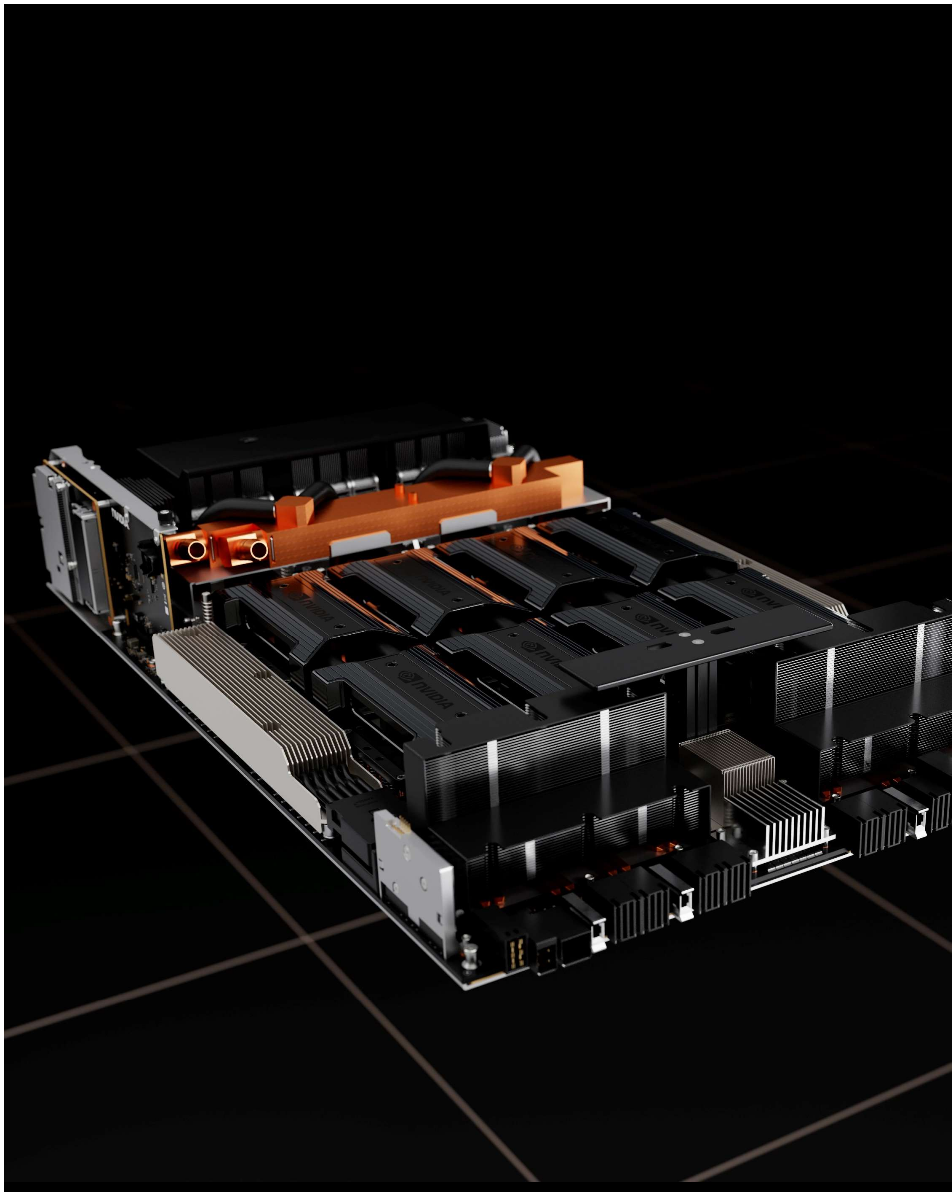


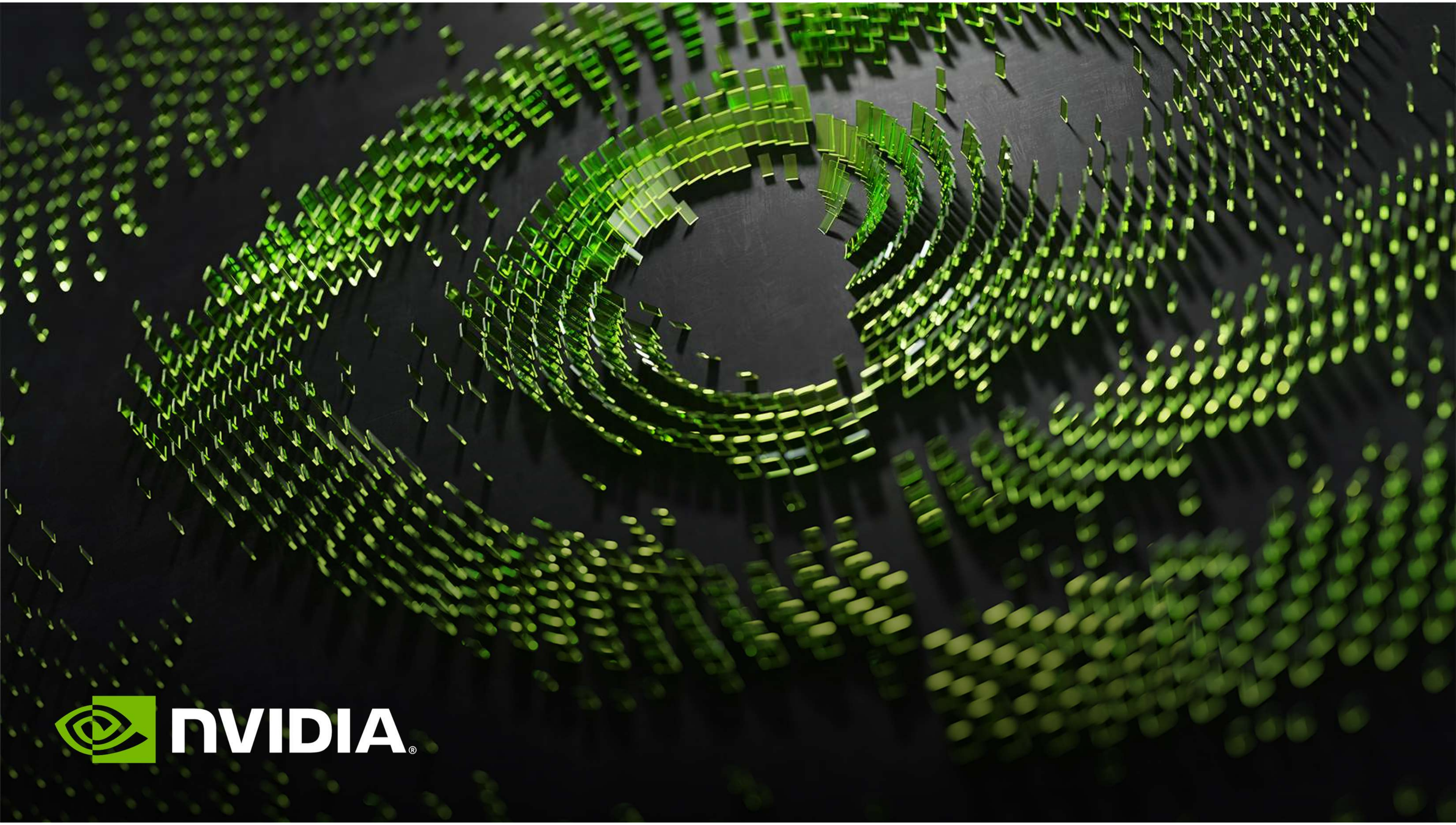
QUESTIONS?

HOPPER DELIVERS A GENERATIONAL LEAP IN PERFORMANCE, EFFICIENCY, AND SECURITY

H100 Whitepaper
www.nvidia.com/hopper-architecture-whitepaper

THANKS TO THE MANY NVIDIA ENGINEERS WHO DESIGNED AND BUILT THE H100 GPU AND THOSE WHO CONTRIBUTED TO THIS PRESENTATION





NVIDIA®