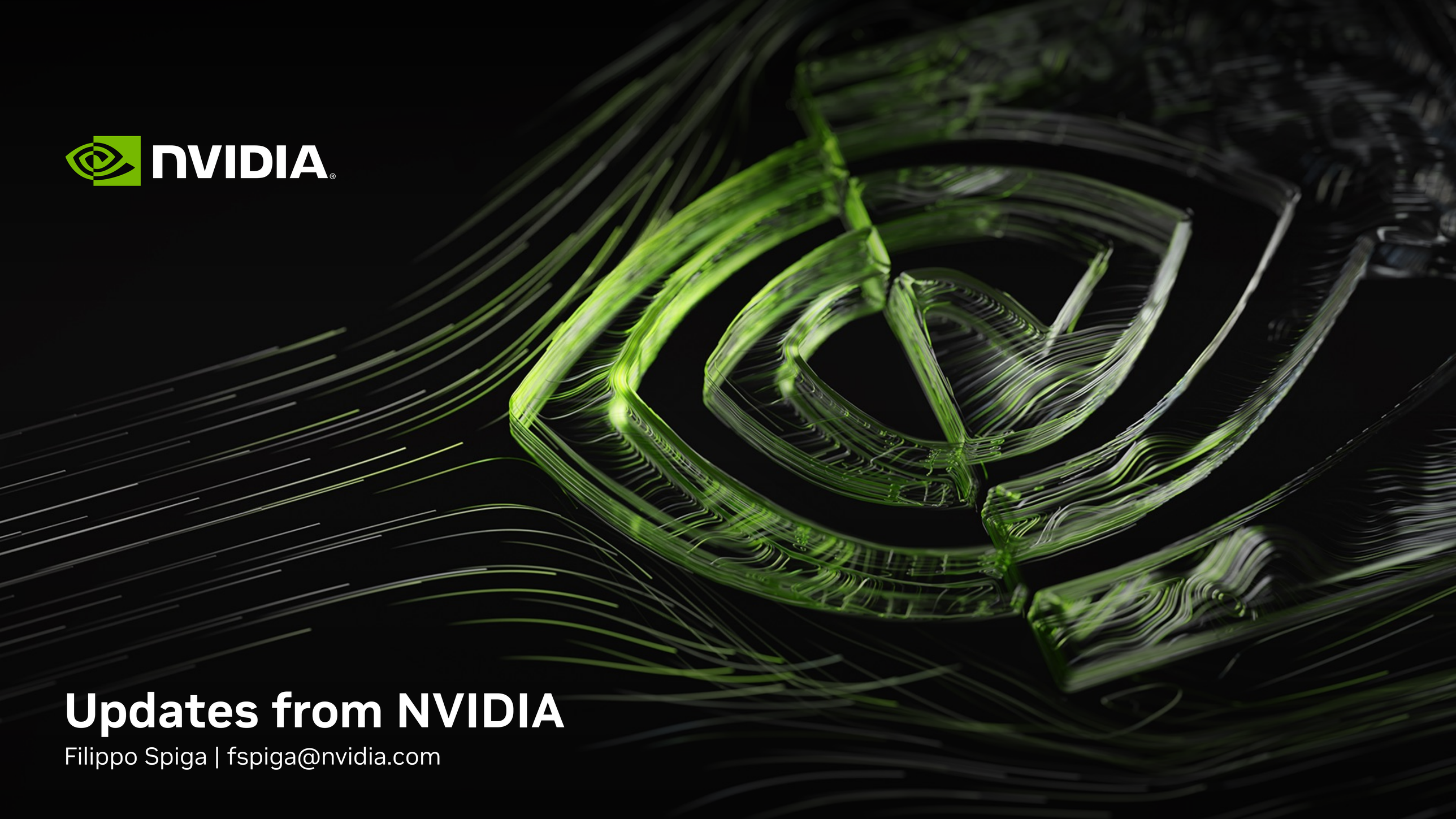


# Updates from NVIDIA

Filippo Spiga | fspiga@nvidia.com

THIS INFORMATION IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. MANY OF THE PRODUCTS AND FEATURES DESCRIBED HEREIN REMAIN IN VARIOUS STAGES AND WILL BE OFFERED ON A WHEN-AND-IF-AVAILABLE BASIS. THIS ROADMAP DOES NOT CONSTITUTE A COMMITMENT, PROMISE, OR LEGAL OBLIGATION AND IS SUBJECT TO CHANGE AT THE SOLE DISCRETION OF NVIDIA. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITIES DESCRIBED FOR OUR PRODUCTS REMAINS AT THE SOLE DISCRETION OF NVIDIA. NVIDIA WILL HAVE NO LIABILITY FOR FAILURE TO DELIVER OR DELAY IN THE DELIVERY OF ANY OF THE PRODUCTS, FEATURES, OR FUNCTIONS SET FORTH IN THIS DOCUMENT.

# Contents

- Updates from GTC/Computex 2023
- Co-design: the QUDA example
- NVIDIA Superchip platform

# Updates from GTC/Computex 2023

NVIDIA

APPLICATION FRAMEWORKS


PLATFORM

NVIDIA AI

NVIDIA OMNIVERSE

ACCELERATION LIBRARIES

RTX

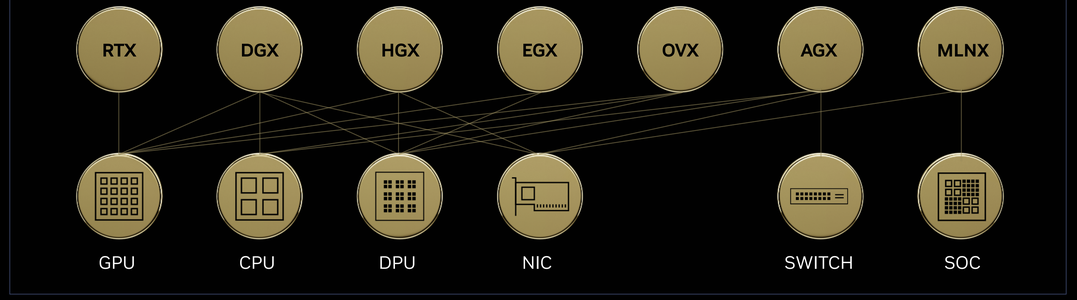
CUDA-X

CUDA

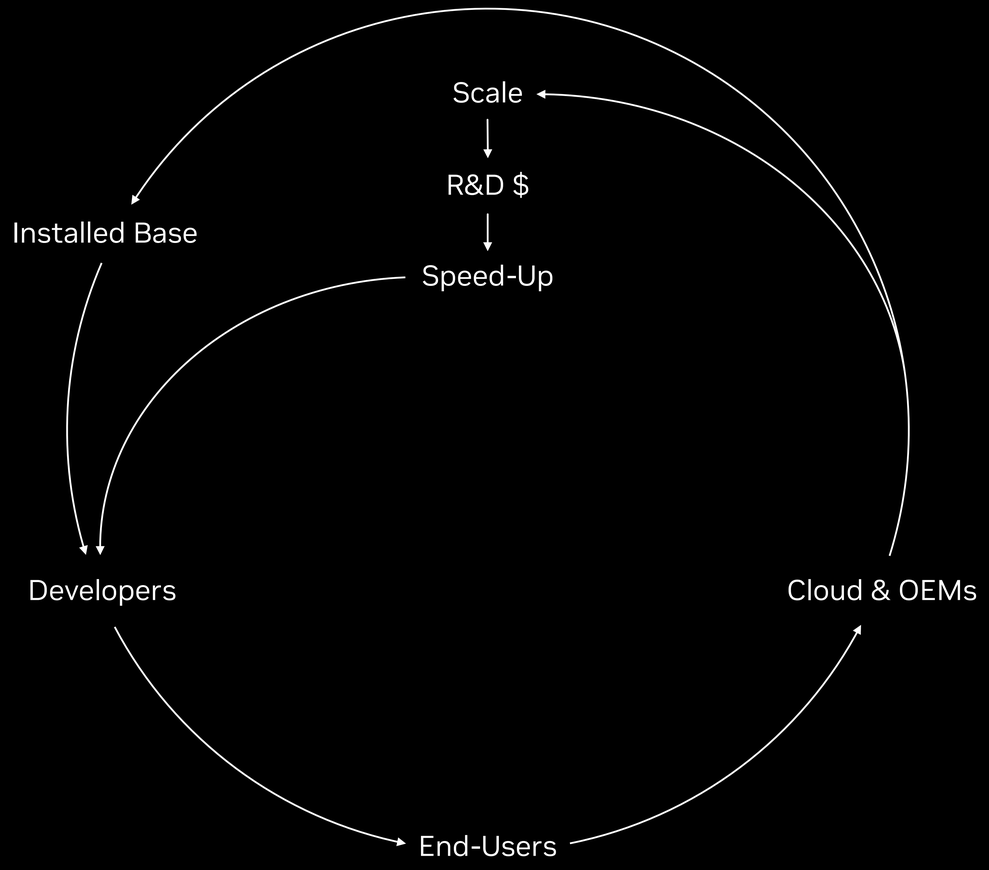
SYSTEM SOFTWARE

Magnum IO | DOCA | Base Command | Forge

HARDWARE



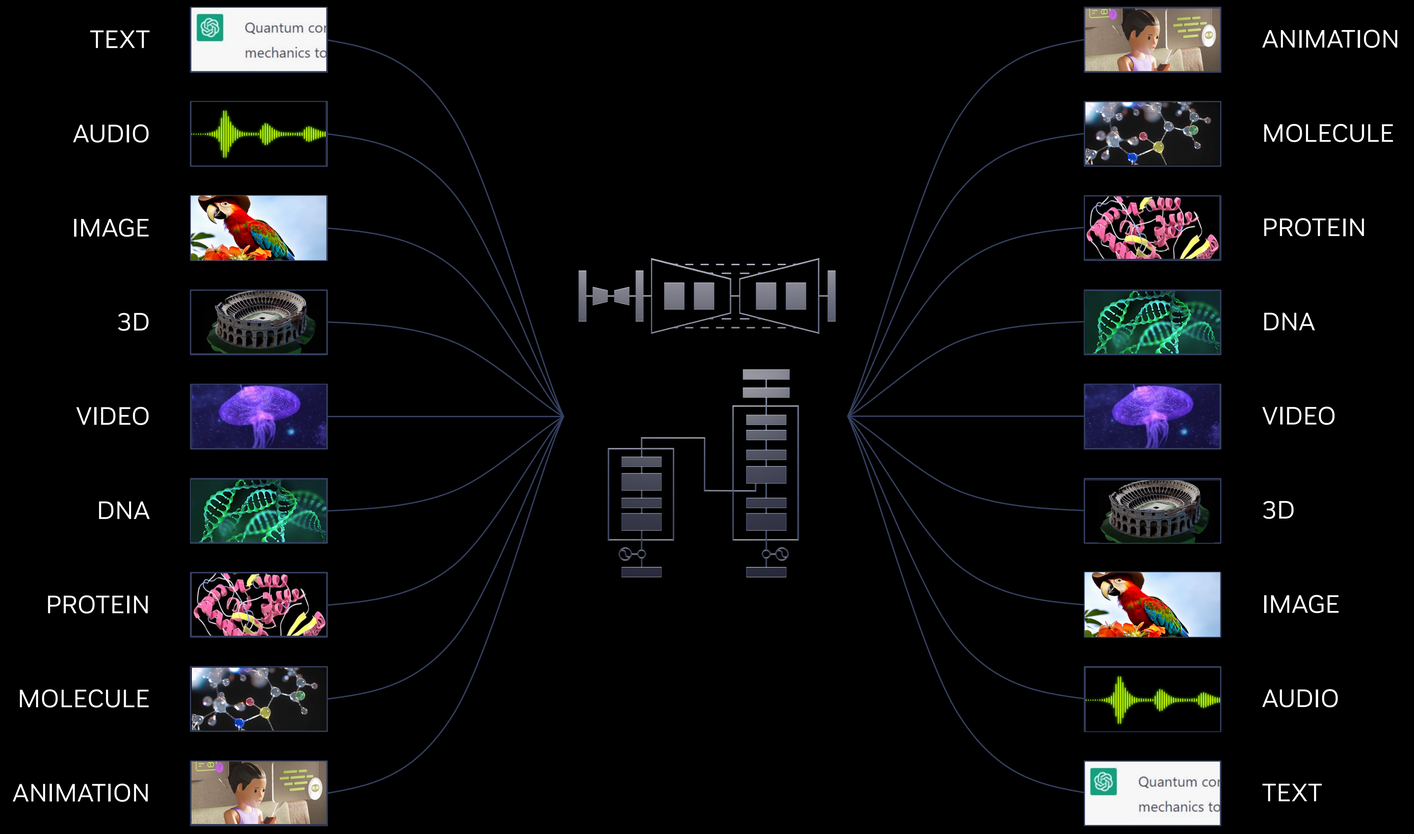
# NVIDIA Accelerated Computing Virtuous Cycle




- 4 Million Developers
- 3000+ Accelerated Applications
- 40 Million+ CUDA Downloads – 25 Million in 2022
- 15,000+ Startups
- 40,000 Enterprises

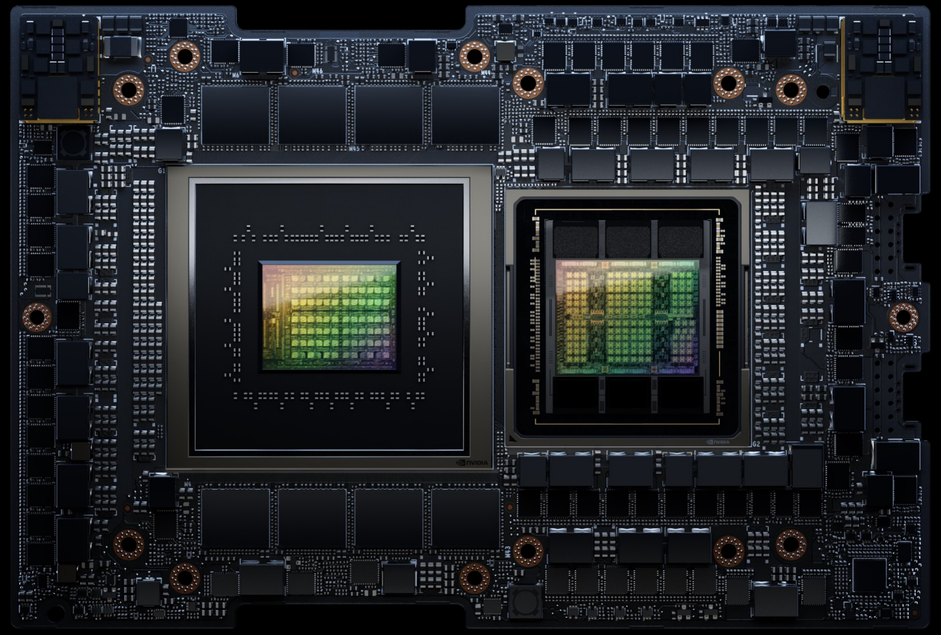
# Generative AI

## The iPhone moment of AI

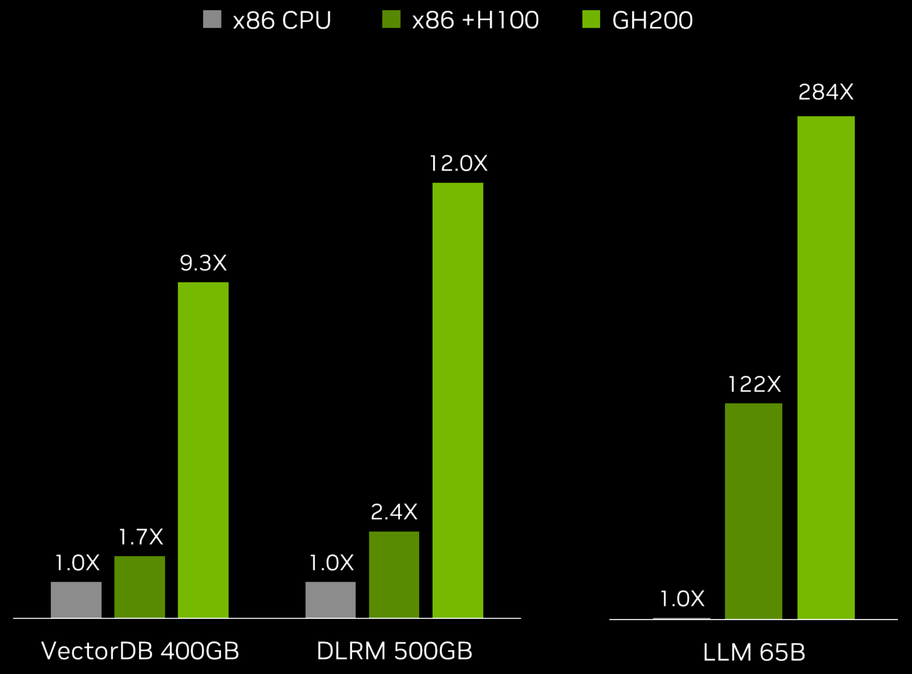


# NVIDIA GH200 Grace Hopper Superchip Now in Production

The engine for the Generative AI era



4 PetaFLOPS TE | 72 Arm CPUs | 96GB HBM3 | 576GB GPU Memory

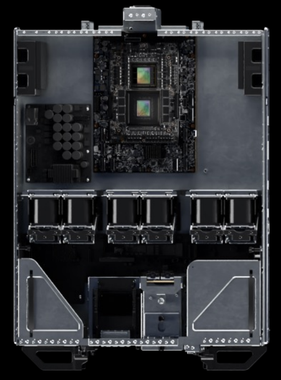


# Exponential Increase in Model Size and Training Data

| 2012 | | 2022 |
| --- | --- | --- |
| AlexNet | | PaLM |
| 62M Parameters | 5,000 X → | 340B Parameters |
| 1.2M Images | 3,000,000 X → | 3.5T Tokens |

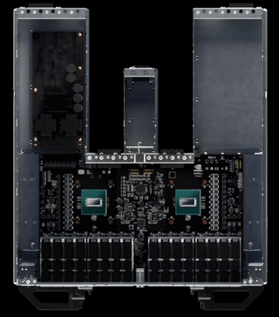
# Introducing DGX GH200

32 X (



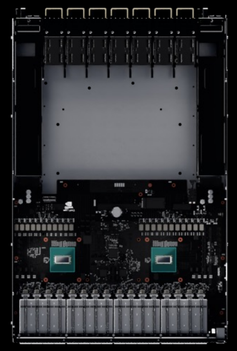
Grace Hopper Superchip x8

+



NVLINK Switch x3

) +



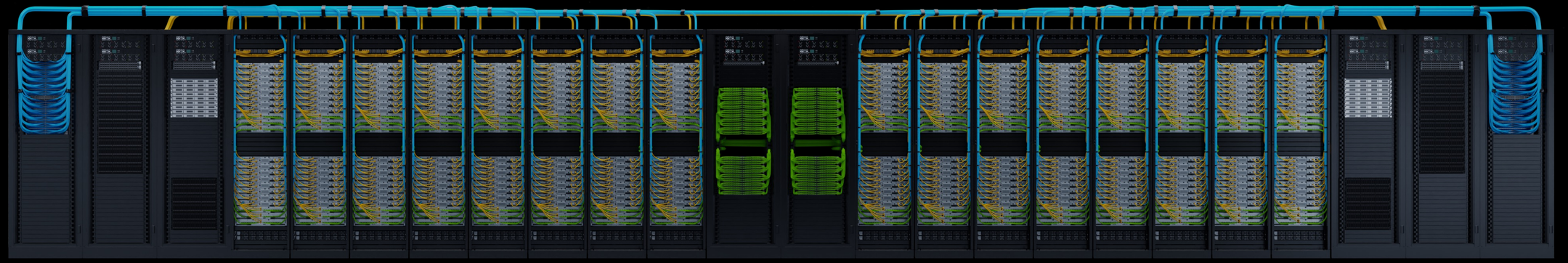
NVLINK Switch x36

=

256 Grace Hopper Superchips

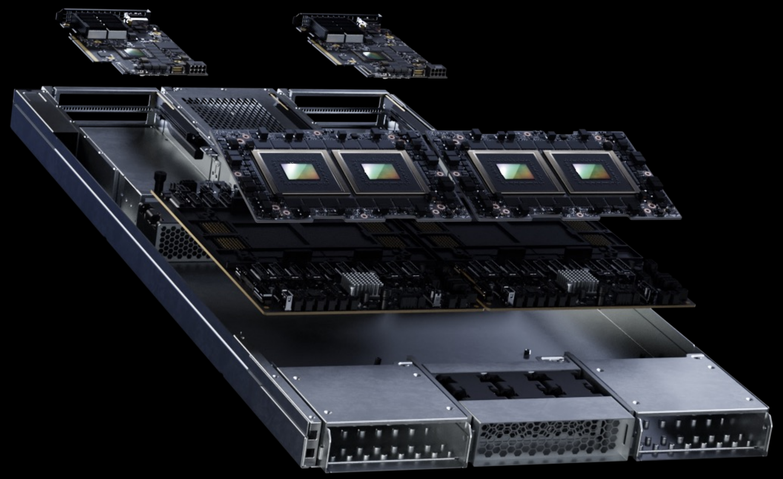
144 TB GPU Memory

150 Miles of Optical Fiber | 2,112 60mm Fans | 70K CFM | 40K lbs | 1 GPU

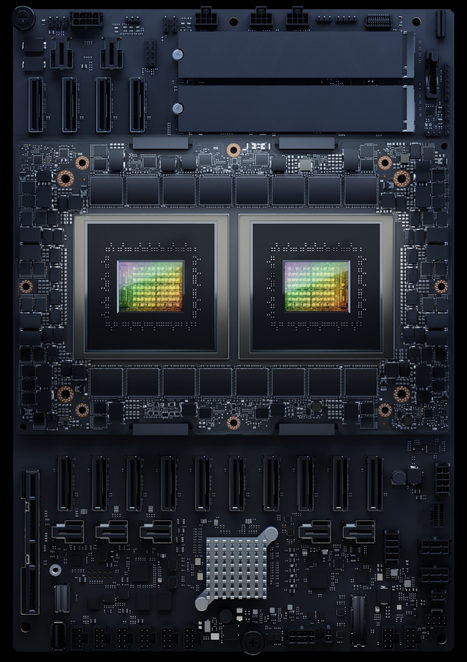


# Dense General-Purpose Grace CPU Server

## Open modular server design for accelerated computing (MGX)



Dense General-Purpose Grace CPU Server



Grace CPU Superchip

**Grace Server**
**580W**

Latest Gen CPU Server
1090W+

Bristol Isambard 3,
LANL Venado, BSC
MareNostrum 5,
Taiwania-4

# Co-design: the QUDA example

Slides extracted by NVIDIA presentation at Lattice 2019
(https://indico.cern.ch/event/764552/)

# GPU COMPUTING FOR LQCD, CIRCA 2009


## Wilson Operator on a GPU (Babich/Barros/Brower/Clark/Osborn/Rebbi)

- Wilson-Operator (GTX 280)
  - Single: 129 Gflops (mat-vec), 110 Gflops (inverter)
  - Double: 39 Gflops (mat-vec), 32 Gflops (inverter)
  - Half: 205 Gflops (mat-vec), 160 Gflops (inverter)
- Wilson-Clover (+5-10% performance)
  - Single: 140 Gflops (mat-vec)
- Algorithms
  - Multi-precision inverter using Reliable Updates (Sleijpen/Van der Vorst)

plenary Beijing 2009

GPUs for LQCD were bleeding edge

~100 GFLOPs per GPU

Single GPU only

Mixed-precision Krylov solvers were the state of the art


# QUDA

- "QCD on CUDA" – http://lattice.github.com/quda (open source, BSD license)
- Effort started at Boston University in 2008, now in wide use as the GPU backend for BQCD, Chroma, CPS, MILC, TIFR, etc.
- Provides:
  - Various solvers for all major fermionic discretizations, with multi-GPU support
  - Additional performance-critical routines needed for gauge-field generation
- Maximize performance
  - Exploit physical symmetries to minimize memory traffic
  - Mixed-precision methods
  - Autotuning for high performance on all CUDA-capable architectures
  - Domain-decomposed (Schwarz) preconditioners for strong scaling
  - Eigenvector and deflated solvers (Lanczos, EigCG, GMRES-DR)
  - Multi-source solvers
  - Multigrid solvers for optimal convergence
- A research tool for how to reach the exascale

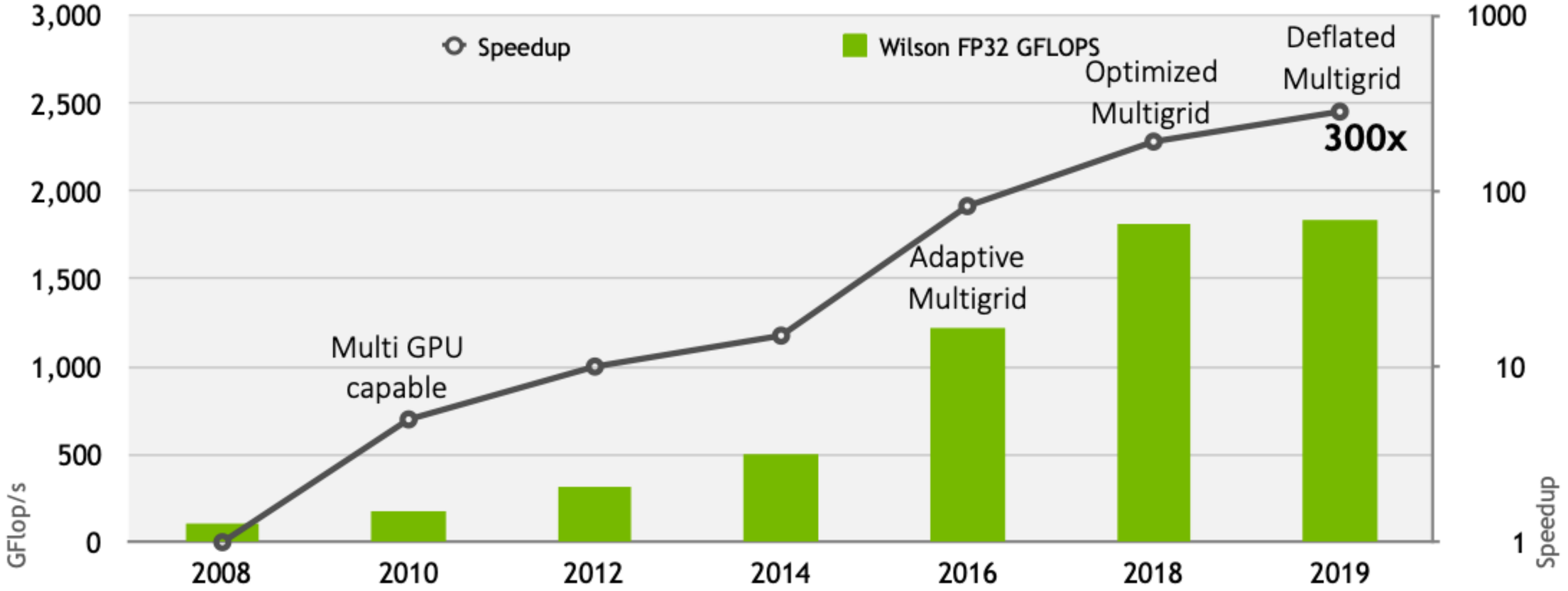
# QUDA CONTRIBUTORS

## 10+ years - lots of contributors

Ron Babich (NVIDIA)
Simone Bacchio (Cyprus)
Kip Barros (LANL)
Rich Brower (Boston University)
Nuno Cardoso (NCSA)
Kate Clark (NVIDIA)
Michael Cheng (Boston University)
Carleton DeTar (Utah University)
Justin Foley (Utah -> NIH)
Joel Giedt (Rensselaer Polytechnic Institute)
Arjun Gambhir (William and Mary)
Steve Gottlieb (Indiana University)
Kyriakos Hadjiyiannakou (Cyprus)
Dean Howarth (BU)
Bálint Joó (Jlab)
Hyung-Jin Kim (BNL -> Samsung)
Bartek Kostrzewa (Bonn)
Claudio Rebbi (Boston University)
Eloy Romero (William and Mary)
Hauke Sandmeyer (Bielefeld)
Guochun Shi (NCSA -> Google)
Mario Schröck (INFN)
Alexei Strelchenko (FNAL)
Jiqun Tu (Columbia)
Alejandro Vaquero (Utah University)
**Mathias Wagner (NVIDIA)**
André Walker-Loud
**Evan Weinberg (NVIDIA)**
Frank Winter (Jlab)

# QUDA NODE PERFORMANCE OVER TIME

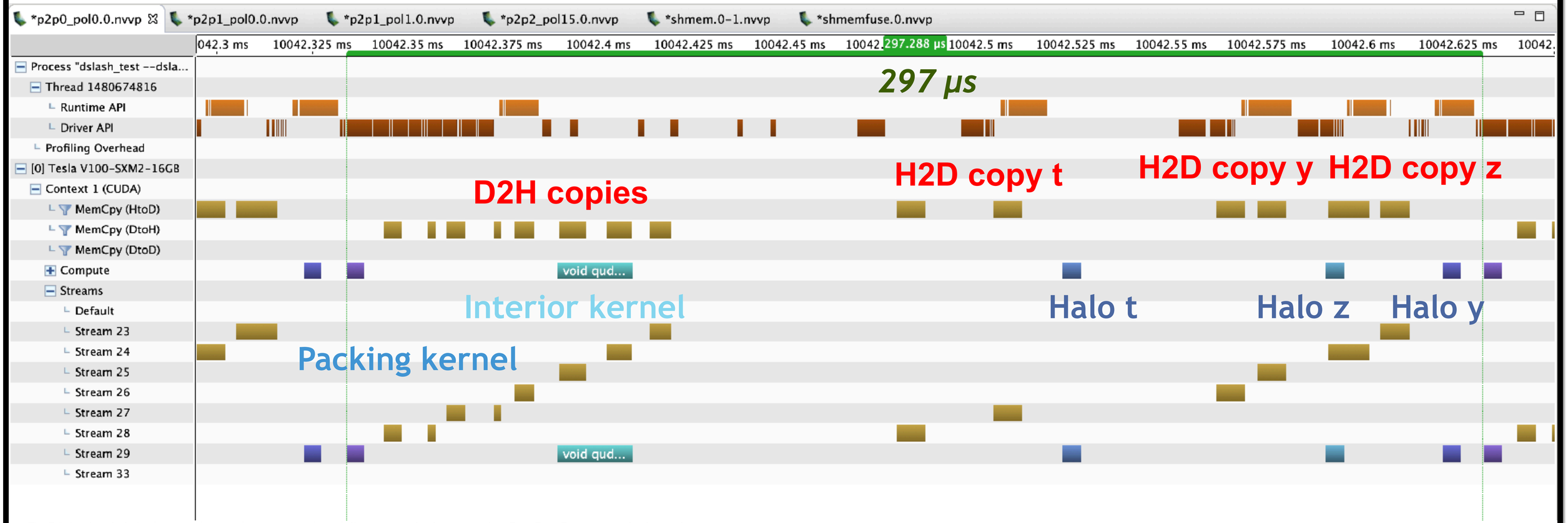
## Multiplicative speedup through software and hardware




Speedup determined by measured time to solution for solving the Wilson operator against a random source on a $V=24^3 64$ lattice, $\beta=5.5$, $M_\pi = 416$ MeV. One node is defined to be 3 GPUs

# WHAT IS LIMITING STRONG SCALING

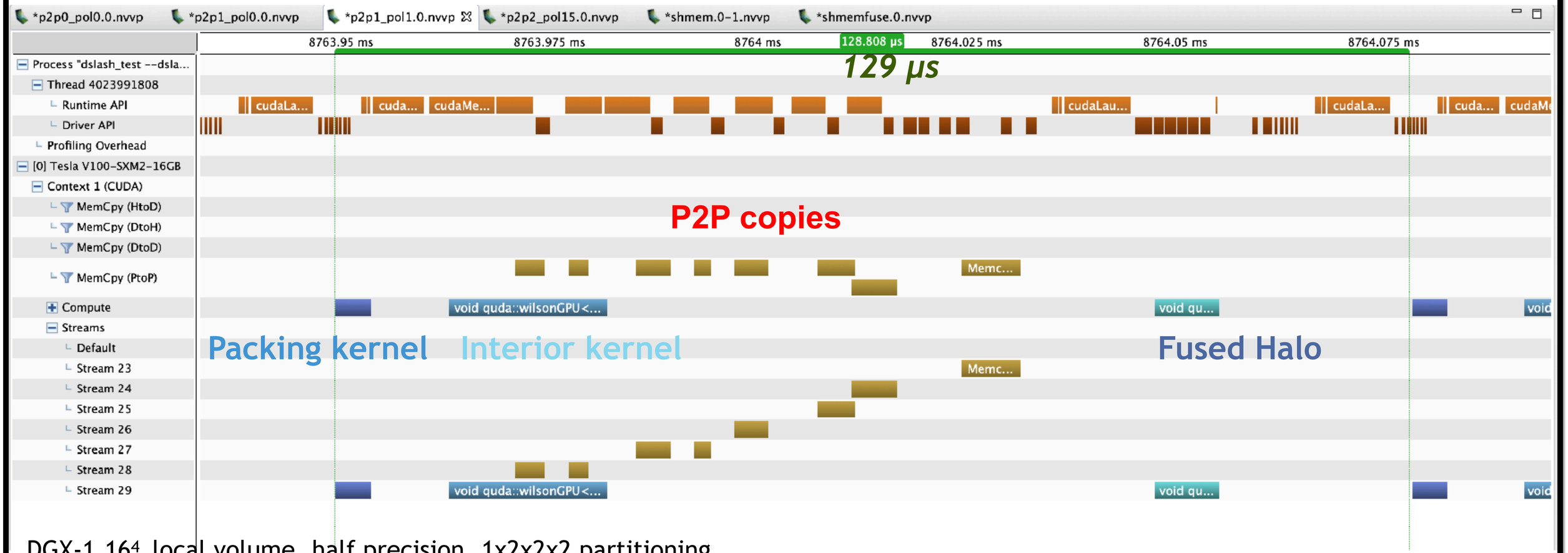
## classical host staging




DGX-1,$16^4$ local volume, half precision, 1x2x2x2 partitioning

# USING NVLINK AND FUSING KERNELS

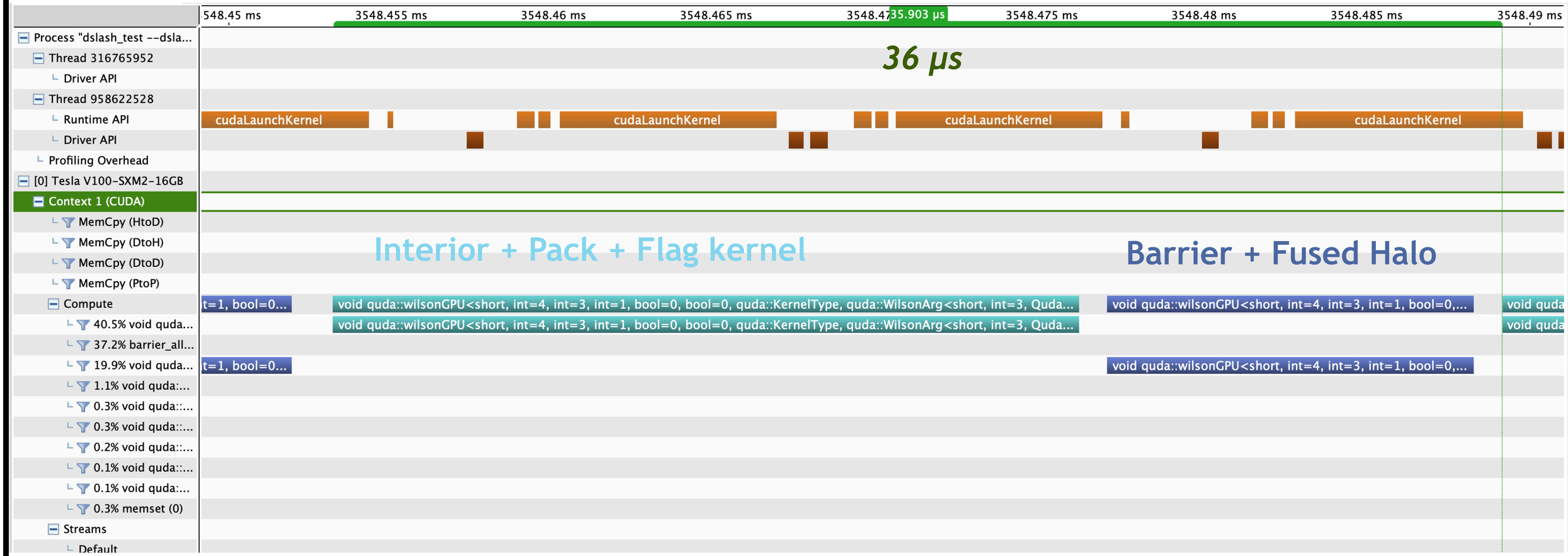
## fewer copies with higher bandwidth, fewer kernels, less API overhead




DGX-1,$16^4$ local volume, half precision, 1x2x2x2 partitioning

# NVSHMEM + FUSING KERNELS

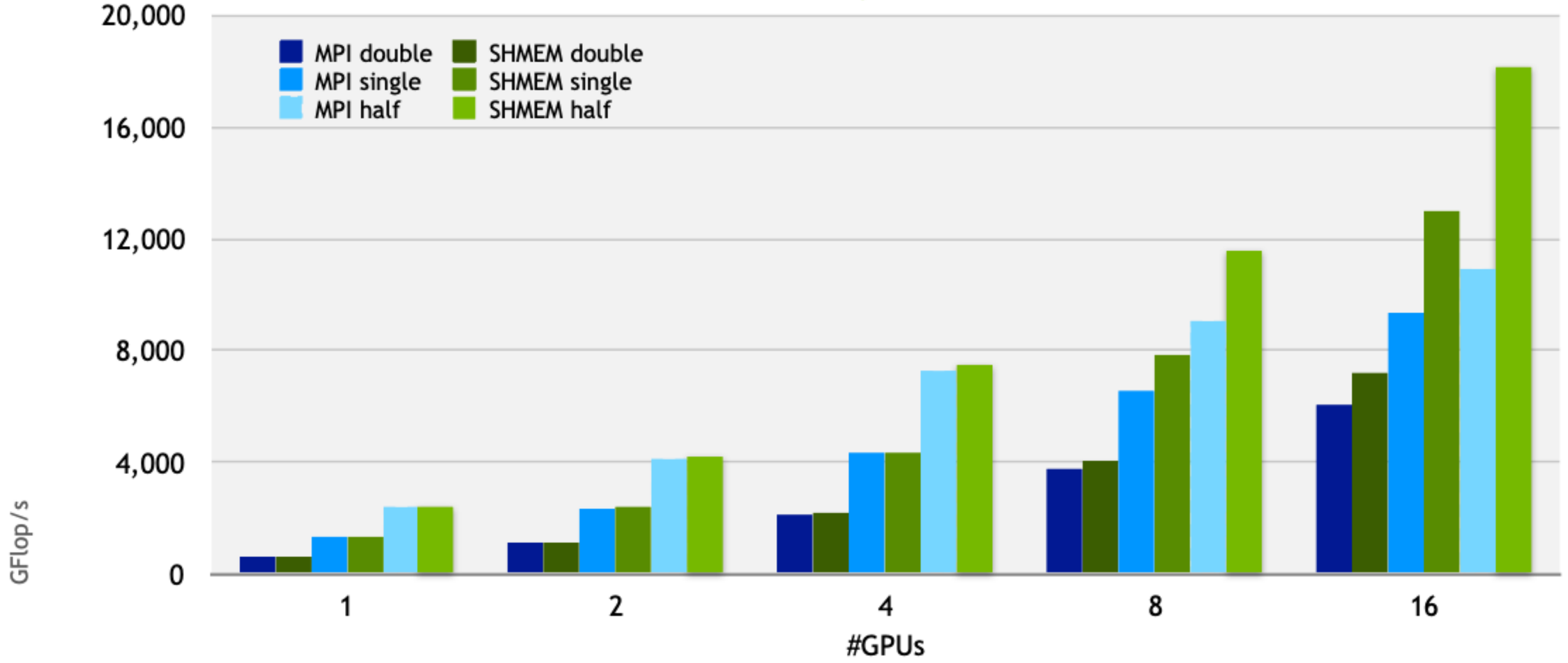
## no extra packing and barrier kernels needed




DGX-1, $16^4$ local volume, half precision, 1x2x2x2 partitioning

# DGX-2 STRONG SCALING

## Global Volume $32^4$, Wilson-Dslash

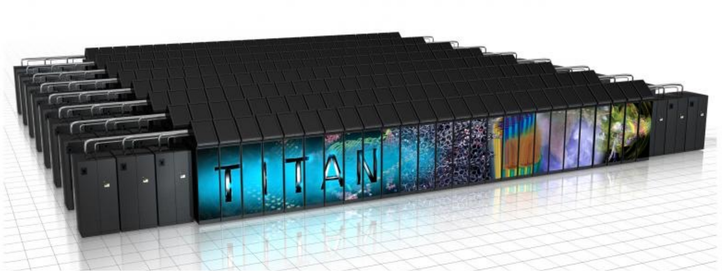


# NVIDIA Superchip platform

# Evolution of GPU accelerated system design

node-level view

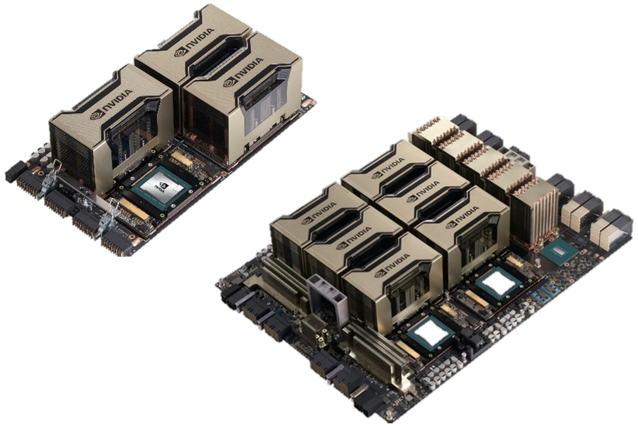
## Single CPU - Single GPU





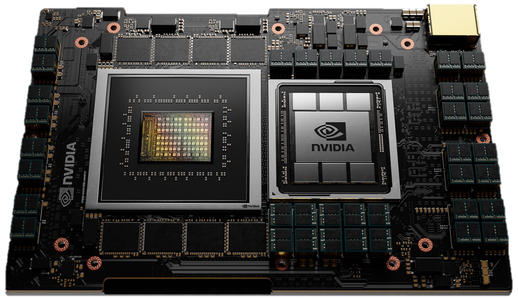
## Single CPU - Multi GPU

("fat" nodes, 2/4/8 ways)


Efficient intra-node GPU-to-GPU, PCIe bottleneck

## Single CPU – Single GPU

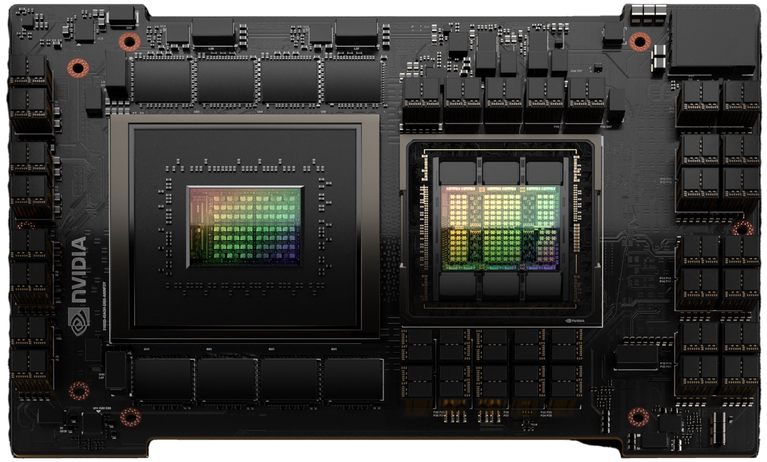



Removing the D2H/H2D bottleneck, adding HW coherency

# NVIDIA Grace for HPC & AI Infrastructure

## Grace Hopper Superchip

Giant Scale AI & HPC



*Accelerated applications where CPU performance and system memory BW are critical; extreme and highly atomic collaboration between CPU & GPU contexts for flagship AI & HPC*

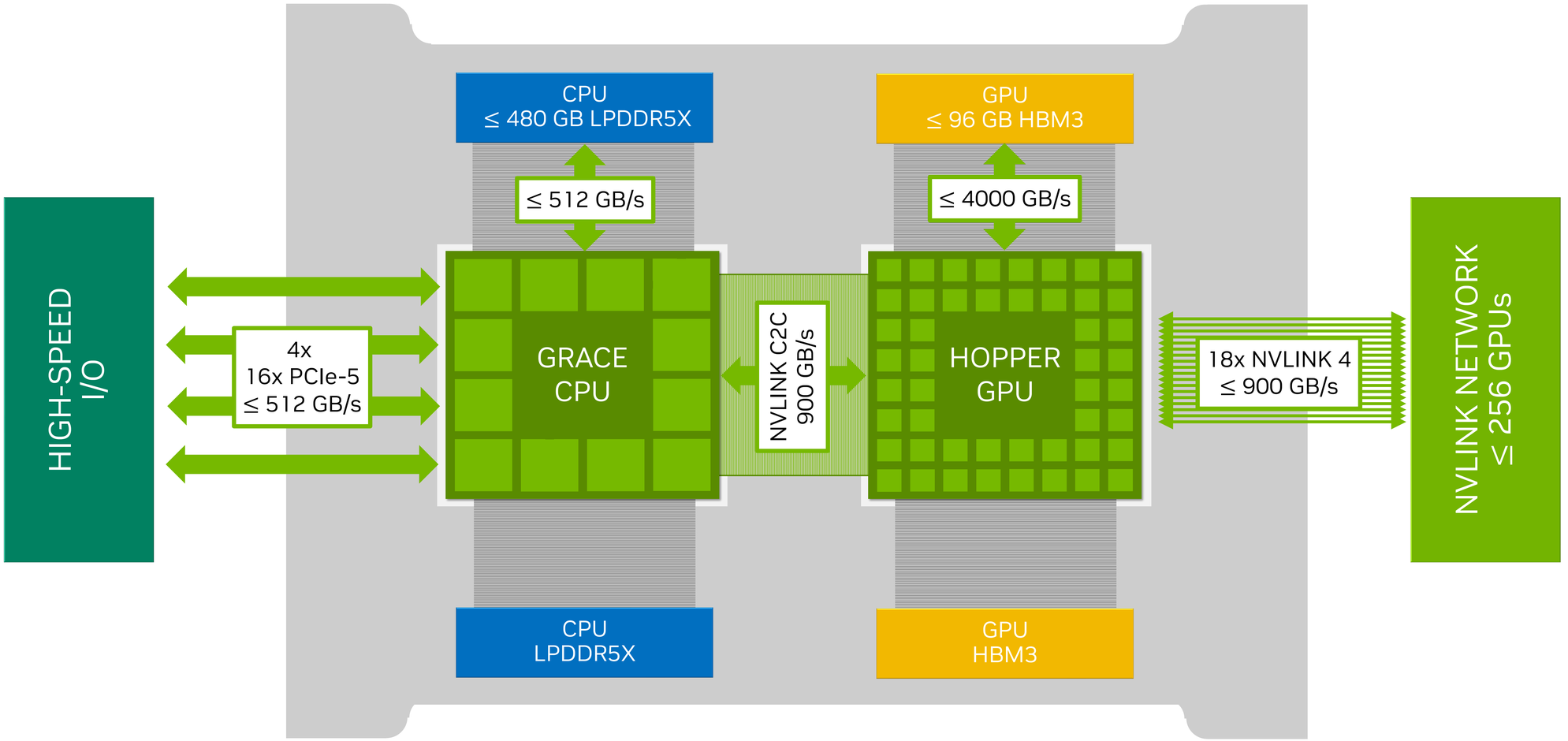
## Grace CPU Superchip

CPU Computing

*Applications that run on CPU but where absolute performance, energy efficiency, and datacenter density matter, such as in scientific computing, data analytics, and hyperscale computing applications*

Designed from the ground-up to be a Superchip, always paired

# Speeds & Feeds



# Programming the NVIDIA platform

CPU, GPU, and Network

## ACCELERATED STANDARD LANGUAGES

ISO C++, ISO Fortran

```
std::transform(par, x, x+n, y, y,
    [=](float x, float y){ return y +
a*x; }
);
```

```
do concurrent (i = 1:n)
   y(i) = y(i) + a*x(i)
enddo
```

```
import cunumeric as np
…
def saxpy(a, x, y):
    y[:] += a*x
```

## INCREMENTAL PORTABLE OPTIMIZATION

OpenACC, OpenMP

```
#pragma acc data copy(x,y) {
...
std::transform(par, x, x+n, y, y,
    [=](float x, float y){
        return y + a*x;
});
...
}
```

```
#pragma omp target data map(x,y) {
...
std::transform(par, x, x+n, y, y,
    [=](float x, float y){
        return y + a*x;
});
...
}
```

## PLATFORM SPECIALIZATION

CUDA

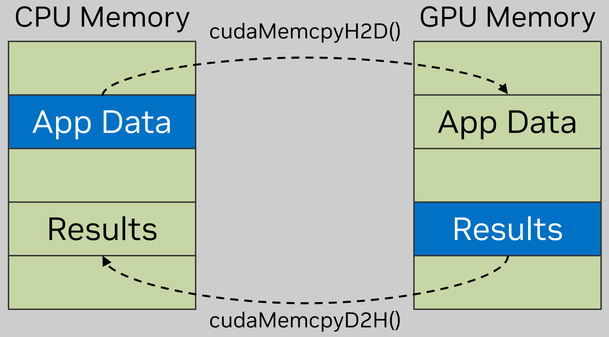
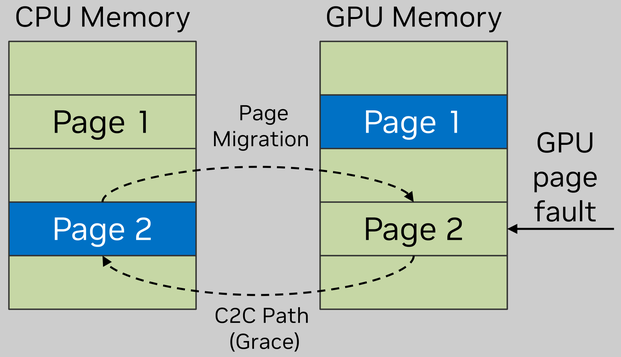
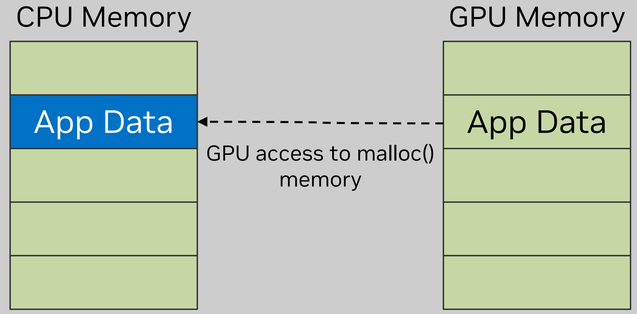
```
__global__
void saxpy(int n, float a,
            float *x, float *y) {
  int i = blockIdx.x*blockDim.x +
          threadIdx.x;
  if (i < n) y[i] += a*x[i];
}

int main(void) {
  ...
  cudaMemcpy(d_x, x, ...);
  cudaMemcpy(d_y, y, ...);

  saxpy<<<(N+255)/256,256>>>(...);

  cudaMemcpy(y, d_y, ...);
```

## ACCELERATION LIBRARIES

| Core | Math | Communication | Data Analytics | AI | Quantum |
|---|---|---|---|---|---|

# ADVANTAGES OF THE GRACE HOPPER MEMORY MODEL

## Full CUDA support with additional Grace memory extensions

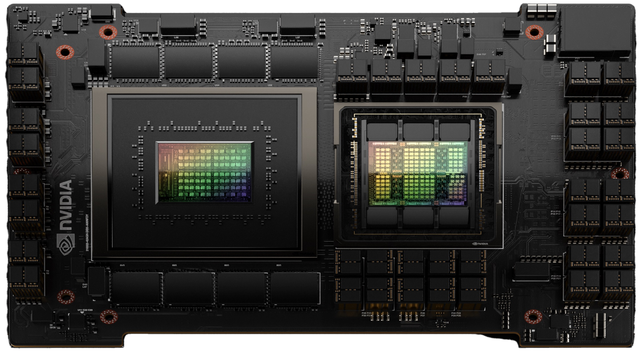
| | Explicit Copy | Managed Memory | System Allocated |
|---|---|---|---|
| | Application explicitly moves data between CPU & GPU as needed | CPU and GPU can access memory on-demand and data migrated locally for higher BW access | GPU can access memory allocated from `malloc()`, `mmap()`, etc. |
| |  |  |  |
| **HGX** | ~60 GB/s PCIe Gen5 transfers (H2D/D2H) | Requires migration to GPU | Access possible with explicit call to `cudaHostRegister()` at PCIe speeds<br>Requires HMM patch in Linux Kernel |
| **G+H** | 7x faster transfers, up to 450 GB/s (NVLink C2C) | Migrations not required and faster migrations when they happen at NVLink C2C speed | `cudaHostRegister()` not needed; access at NVLink C2C speeds |

# Grace Hopper HPC Platform

Unified Memory and Cache Coherence for next gen HPC performance

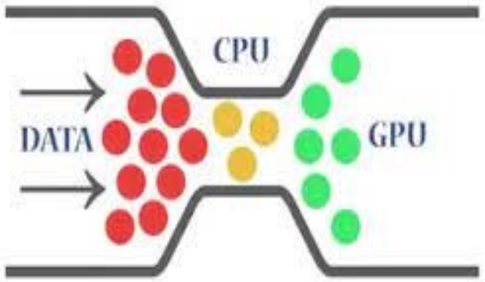
## Partially GPU Accelerated Apps

Big performance gains with no code changes



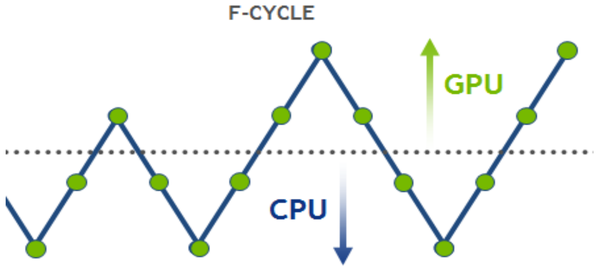

## No More PCIe Bottleneck

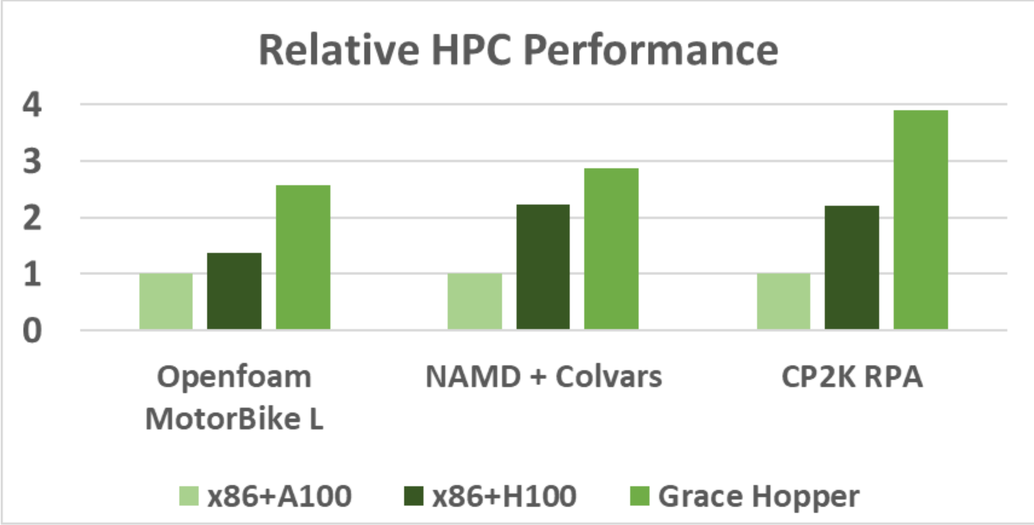
NVLink-C2C is 7X PCIe BW




## CPU & GPU Cache Coherence

Incremental code changes yield big gains




**Fast Access Memory**

600GB

**Memory Bandwidth**

4TB/s

HPC: Preliminary results comparing DGX A100, DGX H100 and Grace Hopper systems.
AI: Simulation results comparing DGX H100 and Grace Hopper systems. GNN: GraphSAGE model for an augmented ogbn- products dataset of 626M nodes and 31B edges, DRLM: Deep Learning Recommendation Model (DLRM) 27TB model, NLP: fine-tuning a GPT-175B model

# Application on Accelerated Systems

## Partially GPU Accelerated

As GPUs become faster applications become increasingly limited by non-GPU factors

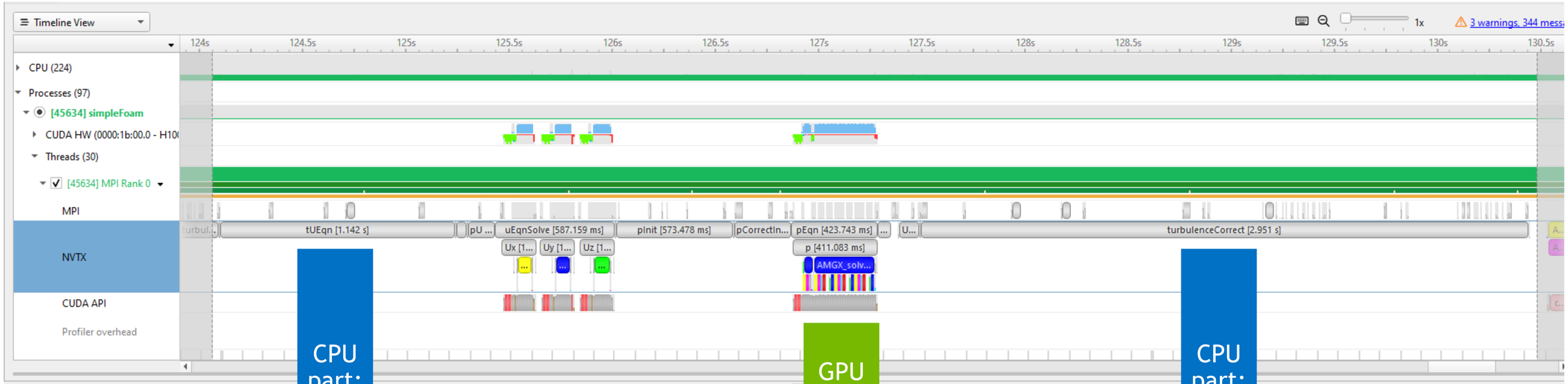
e.g. mostly data transfer (PCIe) limited


- mostly CPU limited

# OpenFoam

## Nsight Systems Profile

x86 + H100 (H100 80GB HBM3)

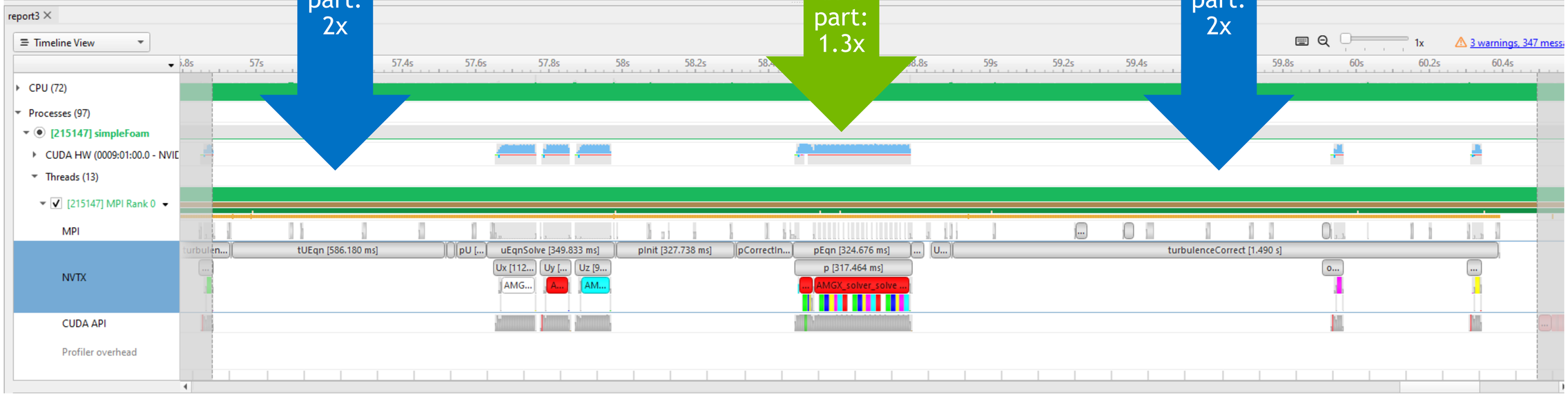



CPU part: 2x

GPU part: 1.3x

CPU part: 2x

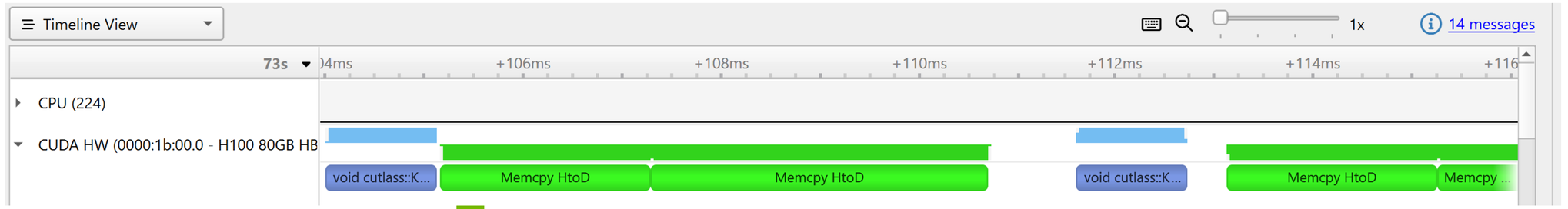
Grace Hopper (H100 96GB HBM3)



# CP2K RPA

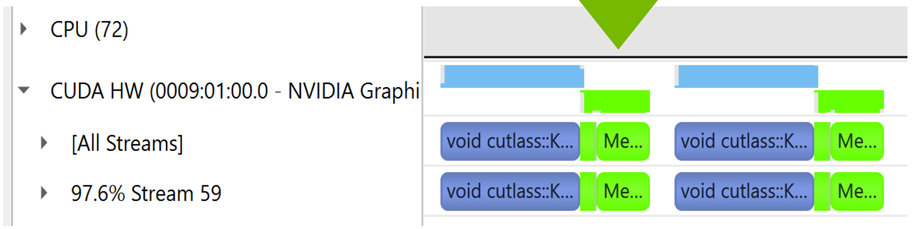
## Nsight Systems Profile

x86 + H100 (H100 80GB HBM3)




**Host to Device 6.2x**

Grace Hopper (H100 96GB HBM3)

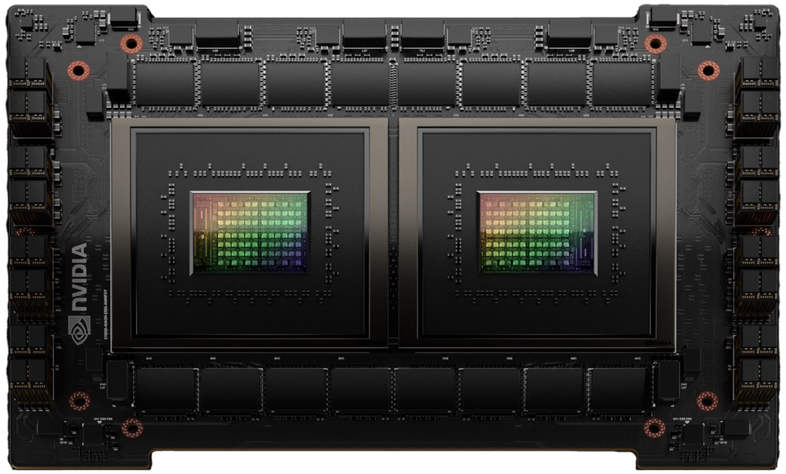



NVIDIA

# FURTHER RESOURCES FOR GRACE CPU AND GRACE HOPPER

## Grace CPU Superchip

- Grace CPU Superchip Architecture Whitepaper
- Grace CPU Architecture In-Depth Blog
- Grace CPU Superchip Data Sheet
- Grace CPU Energy Efficiency Blog
- A Demonstration of AI and HPC Applications for NVIDIA Grace CPU [S51880]




## Grace Hopper Superchip

- Grace Hopper Superchip Architecture Whitepaper
- Grace Hopper Architecture In-Depth Blog
- Grace Hopper Superchip Architecture Data Sheet
- Grace Hopper Recommender System Blog
- Programming Model and Applications for the Grace Hopper Superchip [S51120]