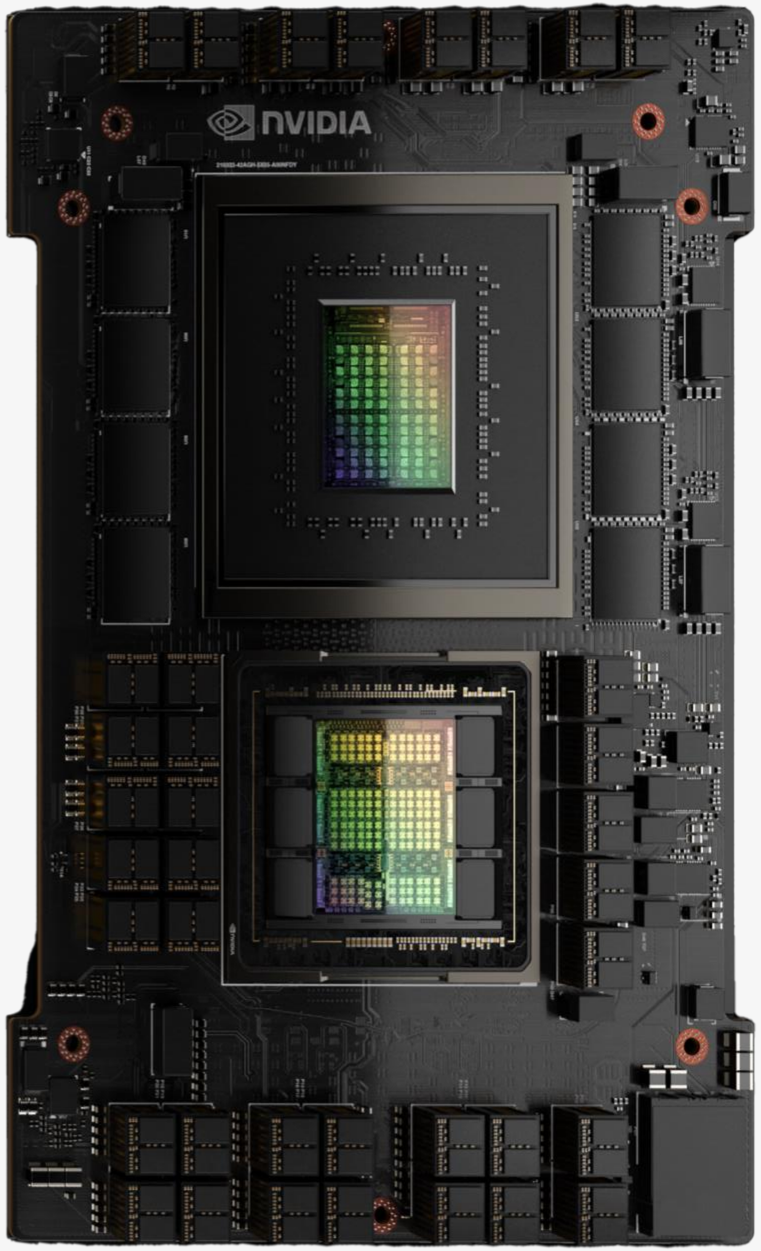
Accelerating Scientific Workflows with the NVIDIA Grace Hopper Platform

Mathias Wagner, Developer Technology Engineer | S62337 | GTC 2024

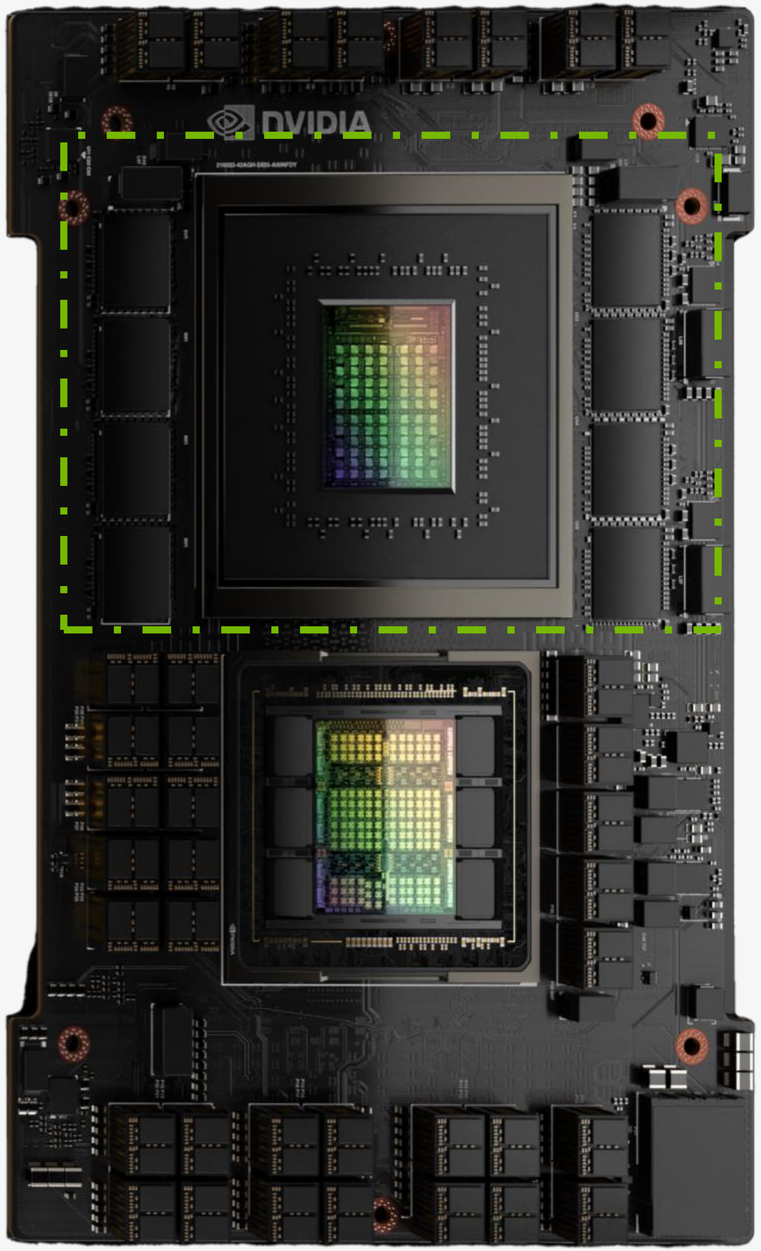


Acknowledgements

Engineers and Scientists made this happen

Listing only direct contributors to this slides … there are many more

- Andre-Walker Loud (LBL)
- Evan Berkowitz (FZ Juelich)
- Andreas Herten (FZ Juelich)
- Lars Hoffmann (FZ Juelich)
- Steve Gottlieb (Indiana University)
- Markus Hrywniak (NVIDIA)
- Jiri Kraus (NVIDIA)
- Matt Martineau (NVIDIA)
- Nikolaos Tselepidis (NVIDIA)
- Dmitry Alexeev (NVIDIA)
- Henry Gu (NVIDIA)
- Alex Chacon (NVIDIA)
- Chris Dallago (NVIDIA)



NVIDIA Grace Hopper Superchip

"super" - more than a "chip"

NVIDIA CPU + NVIDIA GPU w/o compromises

- **NVIDIA Grace CPU**
  - 72 Arm-v9 Neoverse V2 CPU cores with SVE2.
    - → Throughput: 3.6 TFLOP/s
  - Memory:
    - → High capacity: ≤ 480 GB LPDDR5X
    - → High System Memory bandwidth: ≤ 500 GB/s

NVIDIA Grace Hopper Superchip

"super" - more than a "chip"

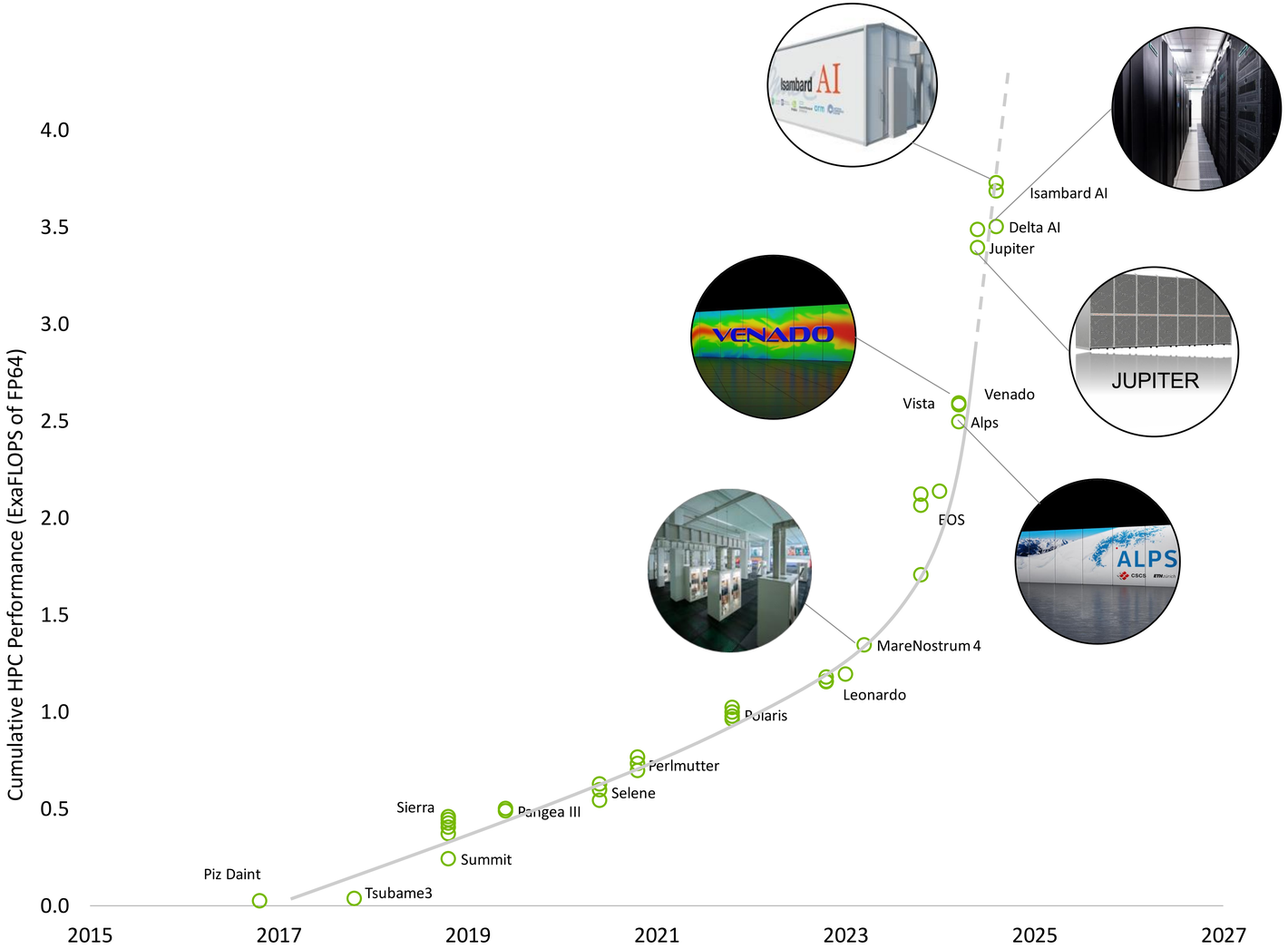
NVIDIA CPU + NVIDIA GPU w/o compromises

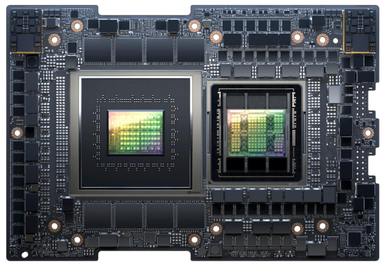
- **NVIDIA Grace CPU**
  - 72 Arm-v9 Neoverse V2 CPU cores with SVE2.
    - → Throughput: 3.6 TFLOP/s
  - Memory:
    - → High capacity: ≤ 480 GB LPDDR5X
    - → High System Memory bandwidth: ≤ 500 GB/s

- **NVIDIA Hopper GPU**
  - → High throughput: 60 TFLOP/s
  - Memory:
    - → Capacity: 96 GB HBM3 / 144 GB HBM3e
    - → Extreme bandwidth ≤ 4000 GB/s / 5000 GB/s
  - ≤ 18x NVLink 4 → 900 GB/s
  - → Threads are threads

Next-Gen Supercomputing Datacenter

4 ExaFLOPs of HPC Performance Driving Scientific Innovation



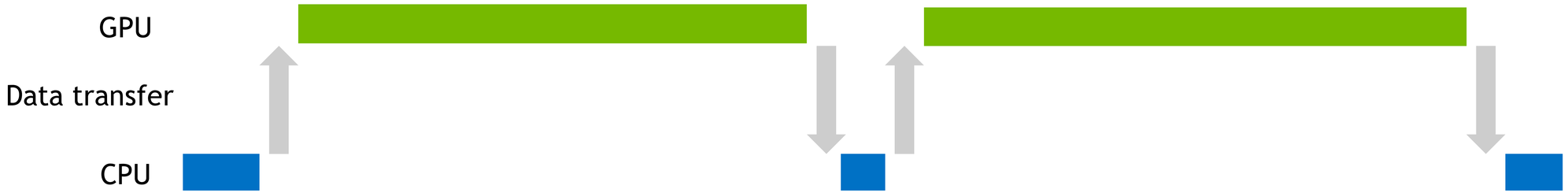


1.7 Exaflops Grace Hopper
Coming online 2024

Application on Accelerated Systems

Fully GPU Accelerated

- Compute almost fully on the GPU with data in GPU memory



- Little to no limitation from CPU and data transfers

Application on Accelerated Systems

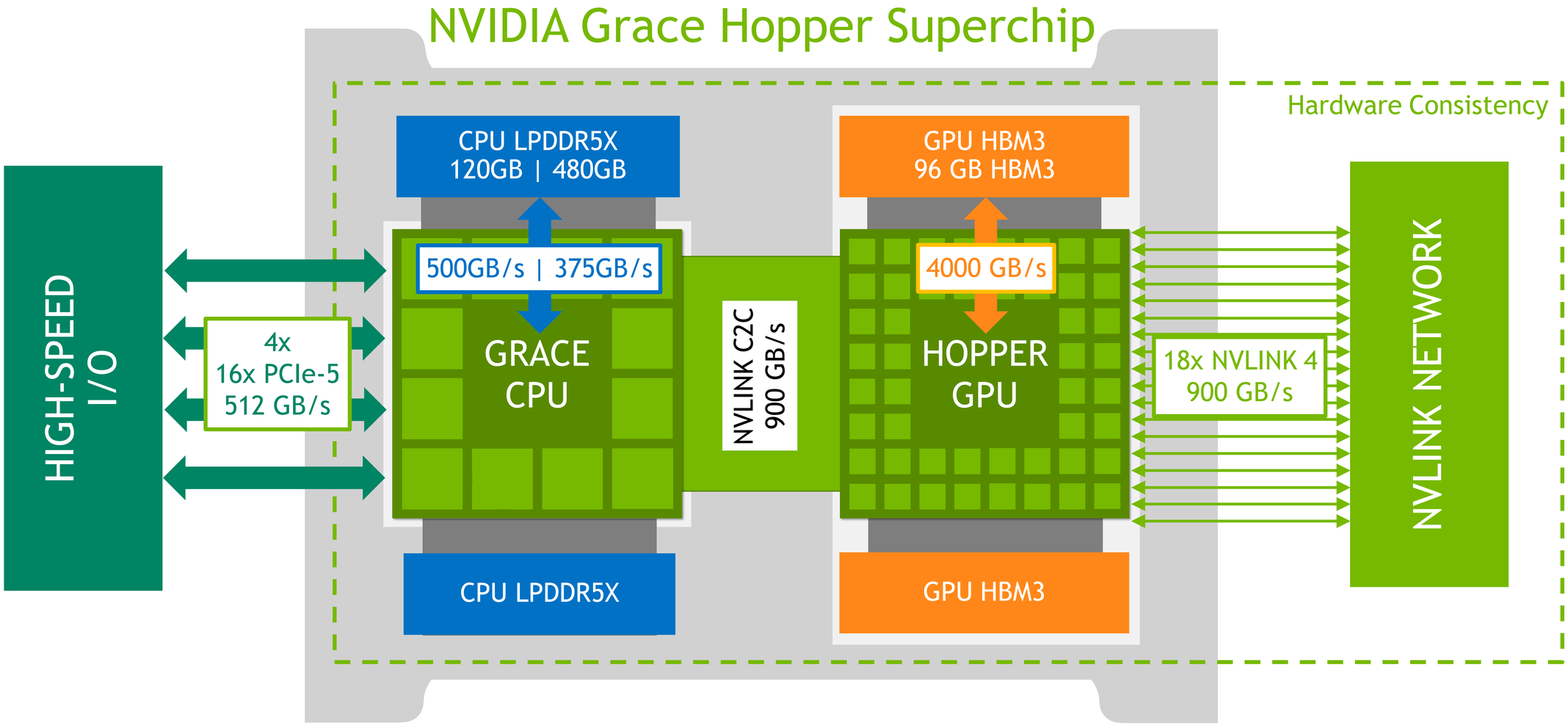
Partially GPU Accelerated

- As GPUs become faster applications become increasingly limited by non-GPU factors, e.g.
- mostly data transfer (PCIe) limited

- mostly CPU limited

Grace Hopper Superchip

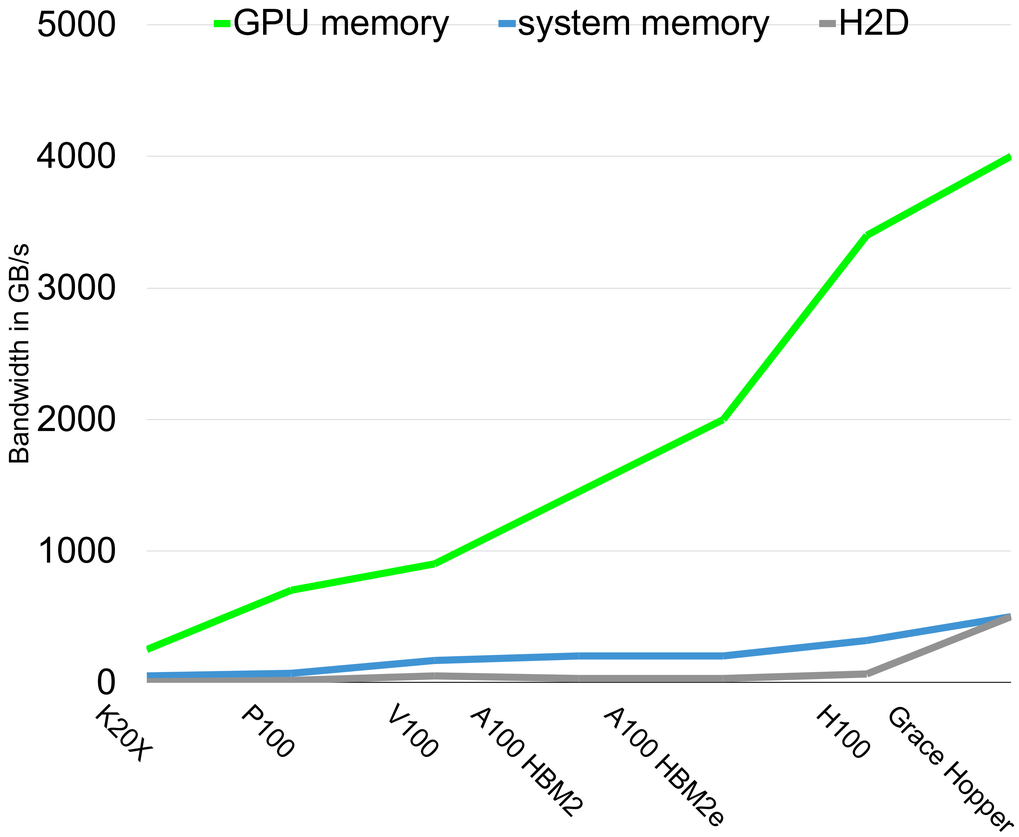
GPU can access CPU memory at CPU memory speeds



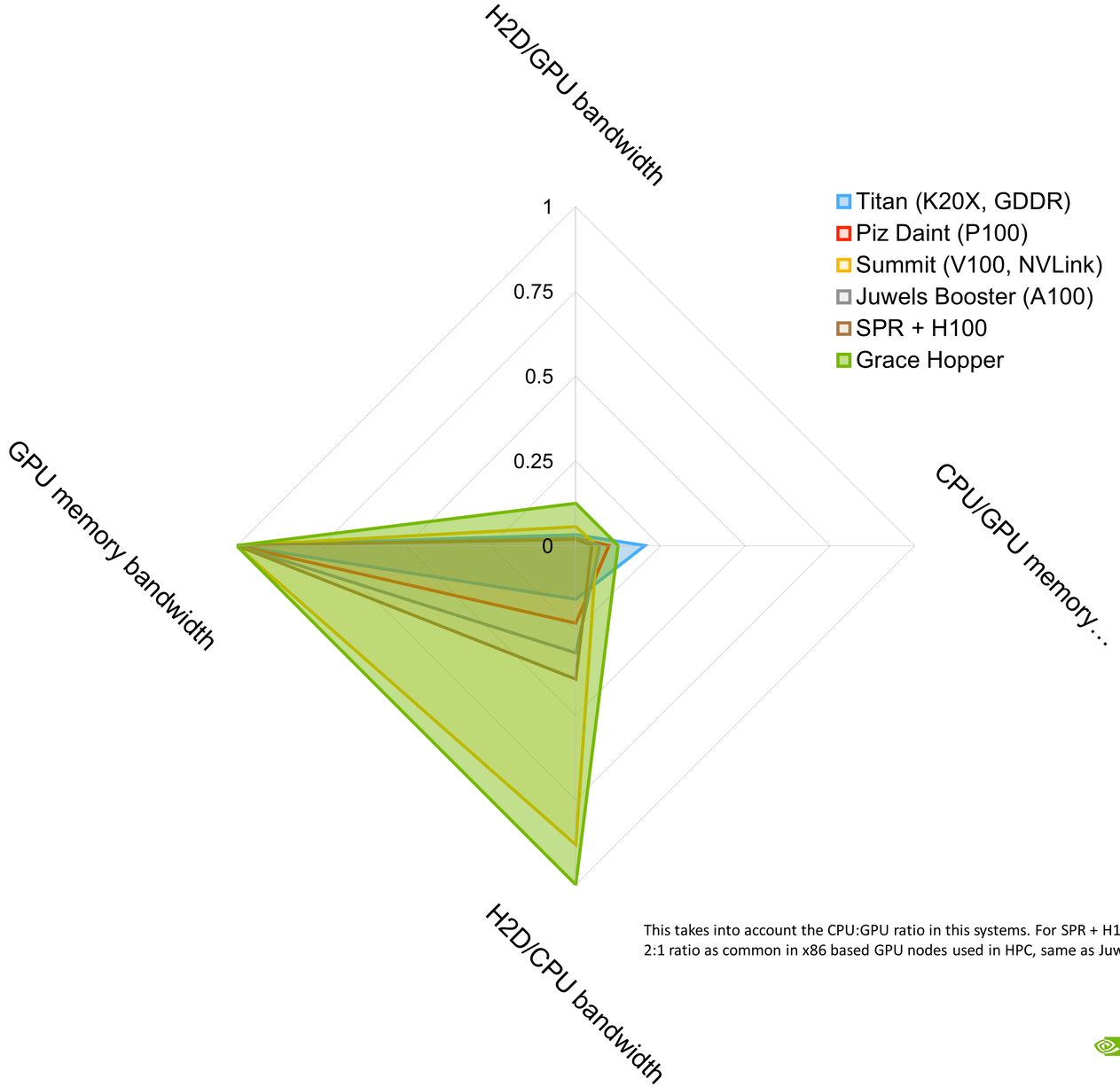
https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper

Widening the bottlenecks

How much do transfer and system memory bandwidth limit your application?



Assumes a typical CPU used in the timeframe.

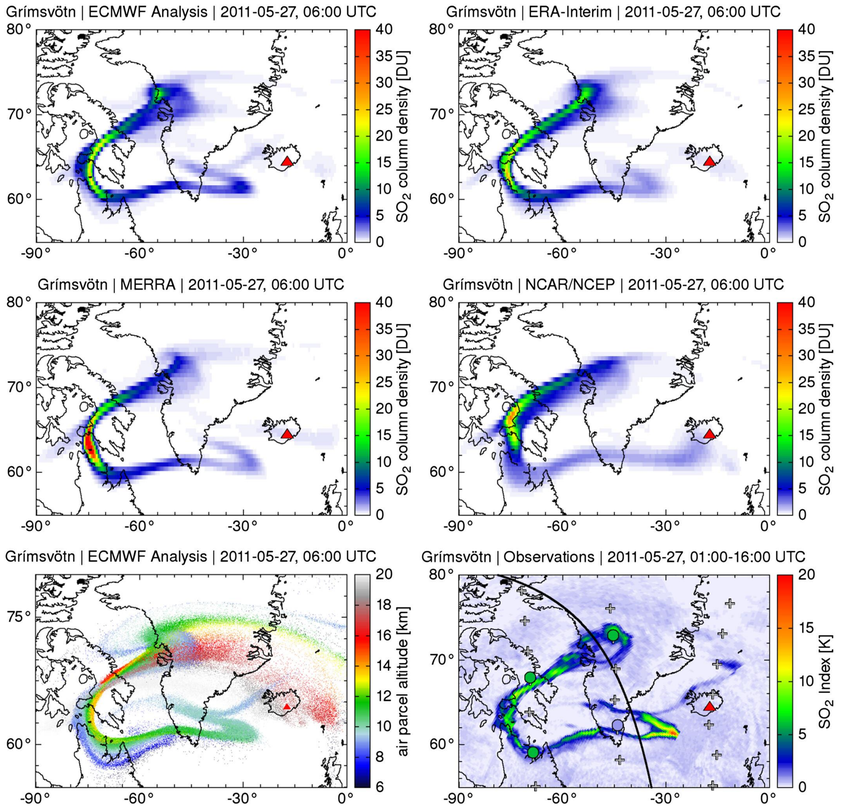


This takes into account the CPU:GPU ratio in this systems. For SPR + H100 we assume a 2:1 ratio as common in x86 based GPU nodes used in HPC, same as Juwels Booster..

Volcanic ash dispersion in the atmosphere

Massive-Parallel Trajectory Calculations (MPTRAC)

- “Lagrangian particle dispersion model for the analysis of atmospheric transport processes in the free troposphere and stratosphere. “
- Allows trajectory calculations of air parcels by solving equations of motion from wind/velocity fields of forecasts or reanalysis data
- Modules for various physical/meteorlogical processes: convection, sedimentation, exponential decay, gas and aqueous phase chemistry, and wet and dry deposition
- Example: Volcanic emissions in the atmosphere following eruption events
  - “Lagrangian transport simulations about 5 days after the eruption of Grímsvötn”
- Details of implemented physics, algorithms at https://slcs-jsc.github.io/mptrac/model-physics/ high



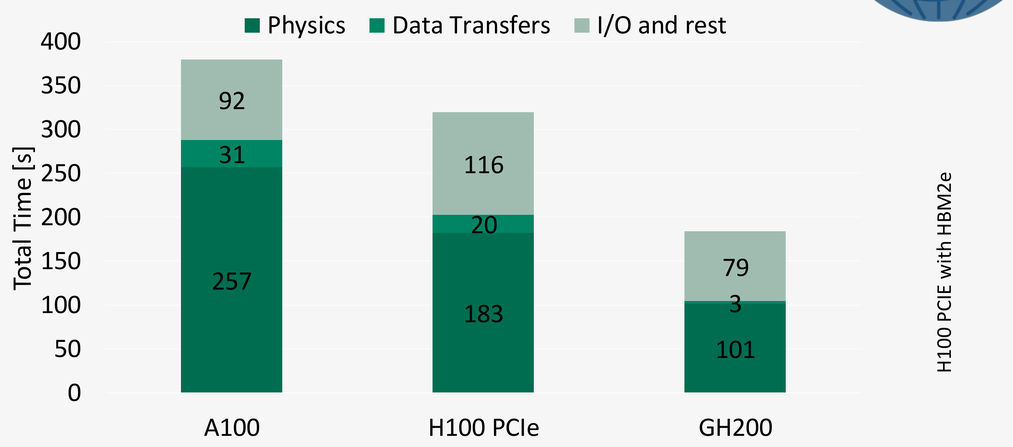
Hoffmann, L., Baumeister, P. F., Cai, Z., Clemens, J., Griessbach, S., Günther, G., Heng, Y., Liu, M., Haghighi Mood, K., Stein, O., Thomas, N., Vogel, B., Wu, X., and Zou, L.: Massive-Parallel Trajectory Calculations version 2.2 (MPTRAC-2.2): Lagrangian transport simulations on graphics processing units (GPUs), Geosci. Model Dev., 15, 2731–2762, https://doi.org/10.5194/gmd-15-2731-2022, 2022.

Hoffmann, L., T. Rößler, S. Griessbach, Y. Heng, and O. Stein, Lagrangian transport simulations of volcanic sulfur dioxide emissions: Impact of meteorological data products, J. Geophys. Res. Atmos., 121, 4651-4673, https://doi.org/10.1002/2015JD023749, 2016.

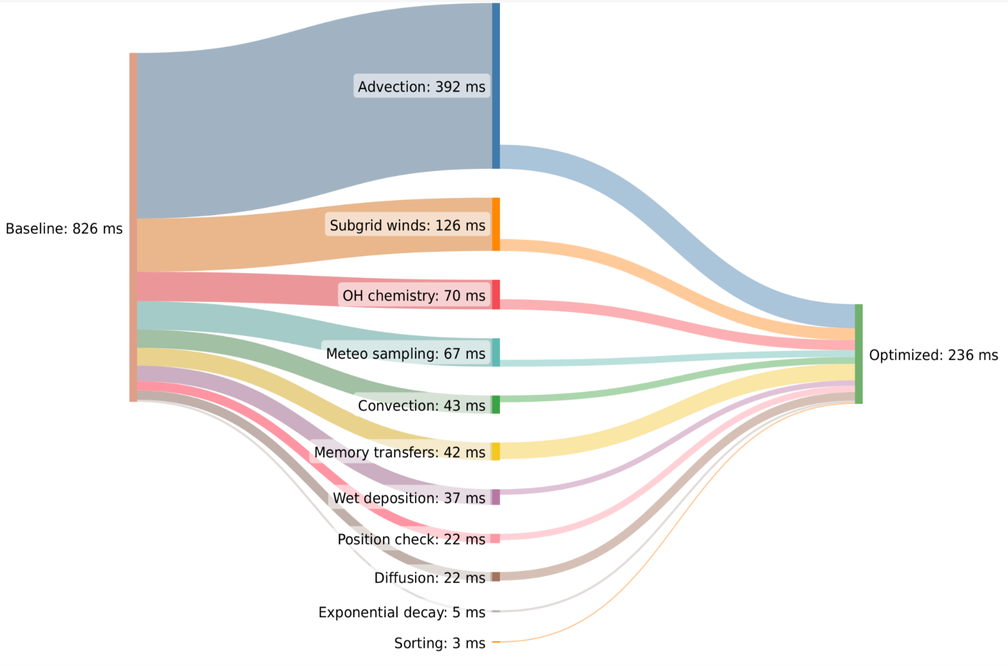
MPTRAC

https://github.com/slcs-jsc/mptrac

- Portable code, but highly optimized implementation
  - OpenMP and OpenACC + MPI parallelization
- Recent performance gains: Improving memory accesses and locality through sorting and improved data layouts.
  Collaboration with JSC, NVIDIA Application Lab [1]
  - Major benefit on GPUs, but large impact on CPU performance as well
  - Measured on JUWELS Booster and JURECA
    - with A100 and H100 GPUs
- Which gains expected on Grace-Hopper?
  - Reading/writing meteorological data from disk: ~flat
  - Transferring data between CPU, GPU



using full CPU socket and 1 GPU



Hoffmann, L., Haghighi Mood, K., Herten, A., Hrywniak, M., Kraus, J., Clemens, J., and Liu, M.: Accelerating Lagrangian transport simulations on graphics processing units: performance optimizations of MPTRAC v2.6, EGUsphere [preprint], https://doi.org/10.5194/egusphere-2023-2547, 2024.

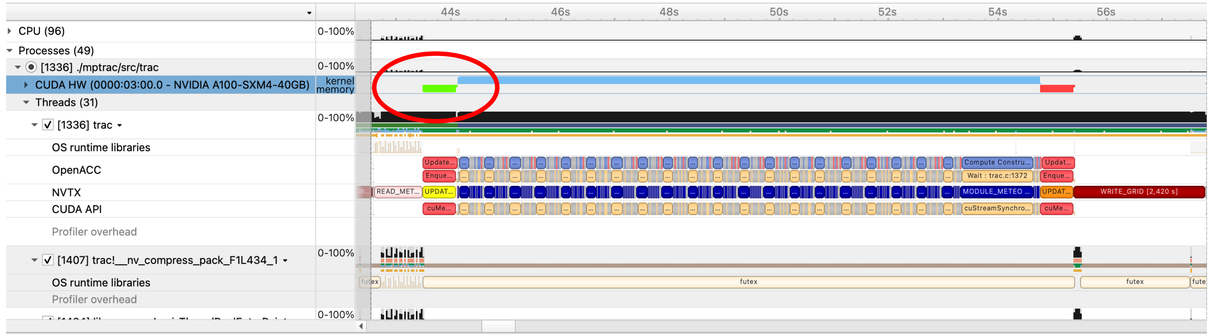
From reference [1]

MPTRAC

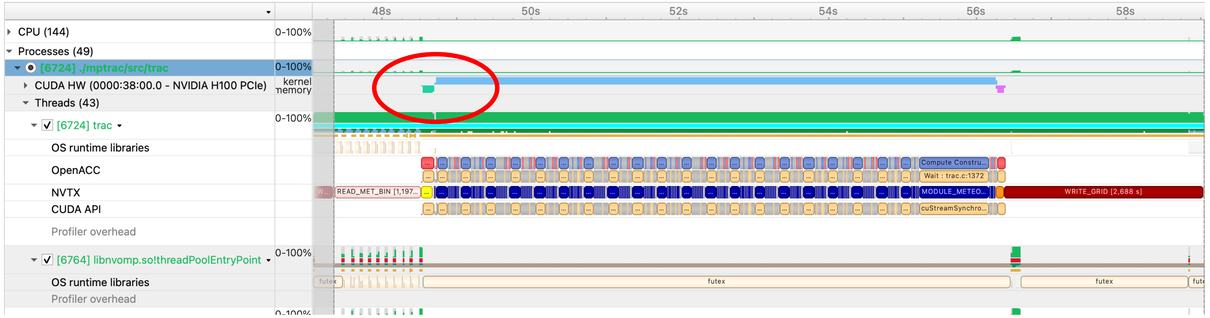
Profile

- Profile comparison (not to scale), excluding I/O:
  - A100 12.0 s
  - H100 PCIe 7.9 s
  - Grace Hopper 4.2 s
- D<->H transfers disappear successively
  - PCIe4 to PCIe5
  - PCIe5 to C2C
- Compounded with GPU generational speedups
  - Faster floating-point throughput
  - Faster HBM
- In progress: Async I/O, overlap with Compute

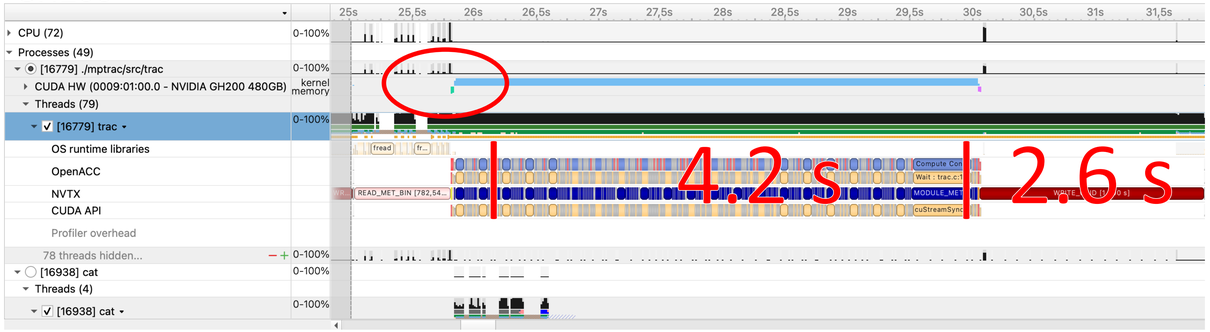
A100



H100



GH200

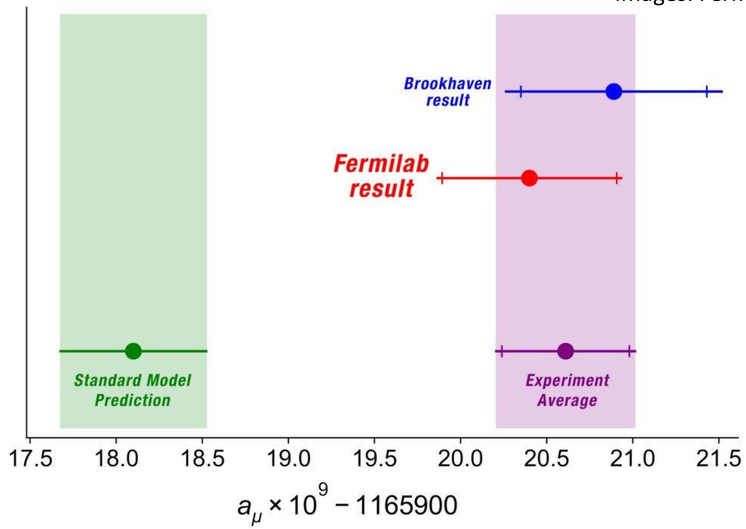


Muon anomalous magnetic moment

New physics?

Images: Fermilab, CERN

- New physics if deviations from Standard Model of Elementary Particle and Nuclear Physics
  - longstanding 3+ standard deviation difference between theory and experiment
  - new experiment was carried out at Fermilab with significantly lower error

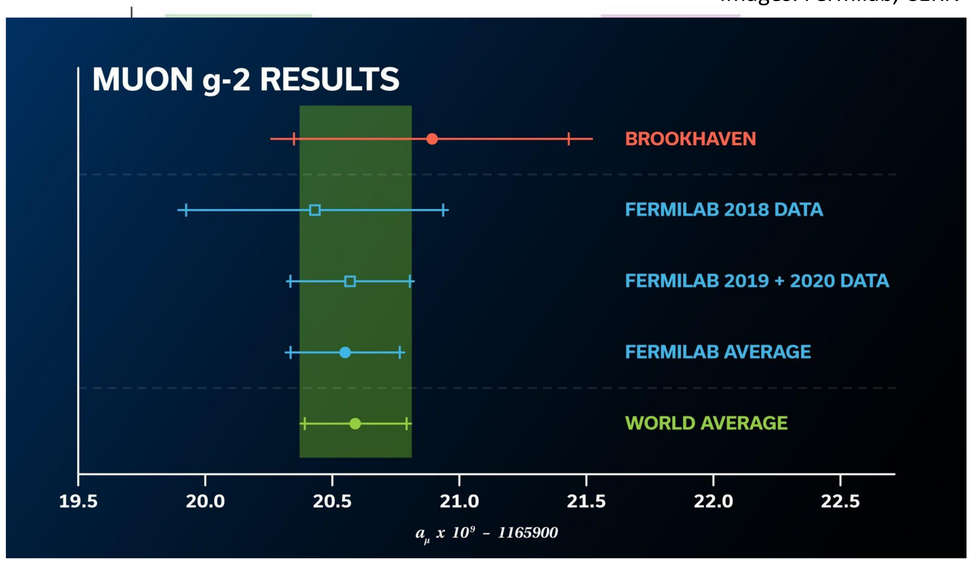


Franz Gross et al., Eur.Phys.J.C 83 (2023), 1125
T. Aoyama et al.Phys.Rept. 887 (2020), 1-166
Fermilab Lattice, HPQCD, and MILC Collaborations, Bazavov et al, Phys.Rev.D 107 (2023) 11, 114514

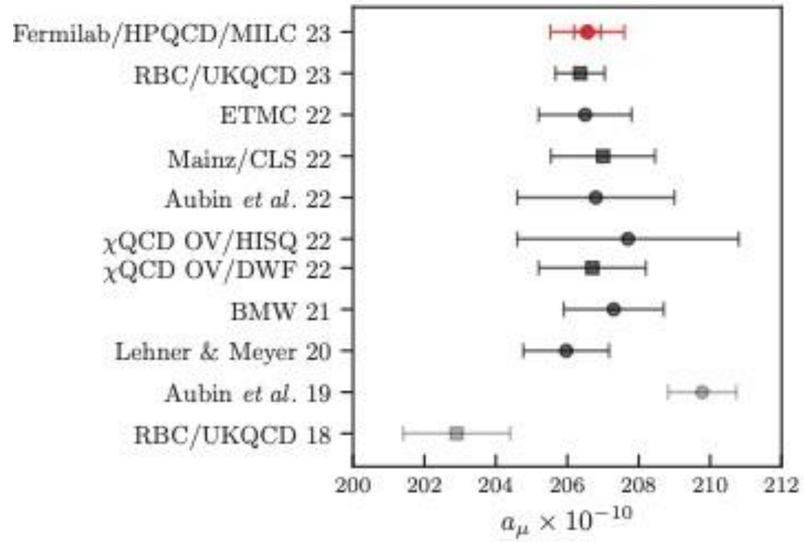
Muon anomalous magnetic moment

New physics?

Images: Fermilab, CERN



- New physics if deviations from Standard Model of Elementary Particle and Nuclear Physics
  - longstanding 3+ standard deviation difference between theory and experiment
  - new experiment was carried out at Fermilab with significantly lower error
- Requires similar precision in theoretical calculations
  - One ab-initio approach is using Lattice QCD
- Quantum Chromodynamics (QCD) is a 50-year-old theory of the strong interaction
  - Interaction between quarks and gluons - which form mesons and baryons
    - E.g. Neutron and Protons
- Lattice QCD is a discretized version
  - Large ensemble to generate enough statistics and fine and larger enough lattices
- Future work will add also electromagnetic properties to simulations
- Future work will also require even finer Lattices

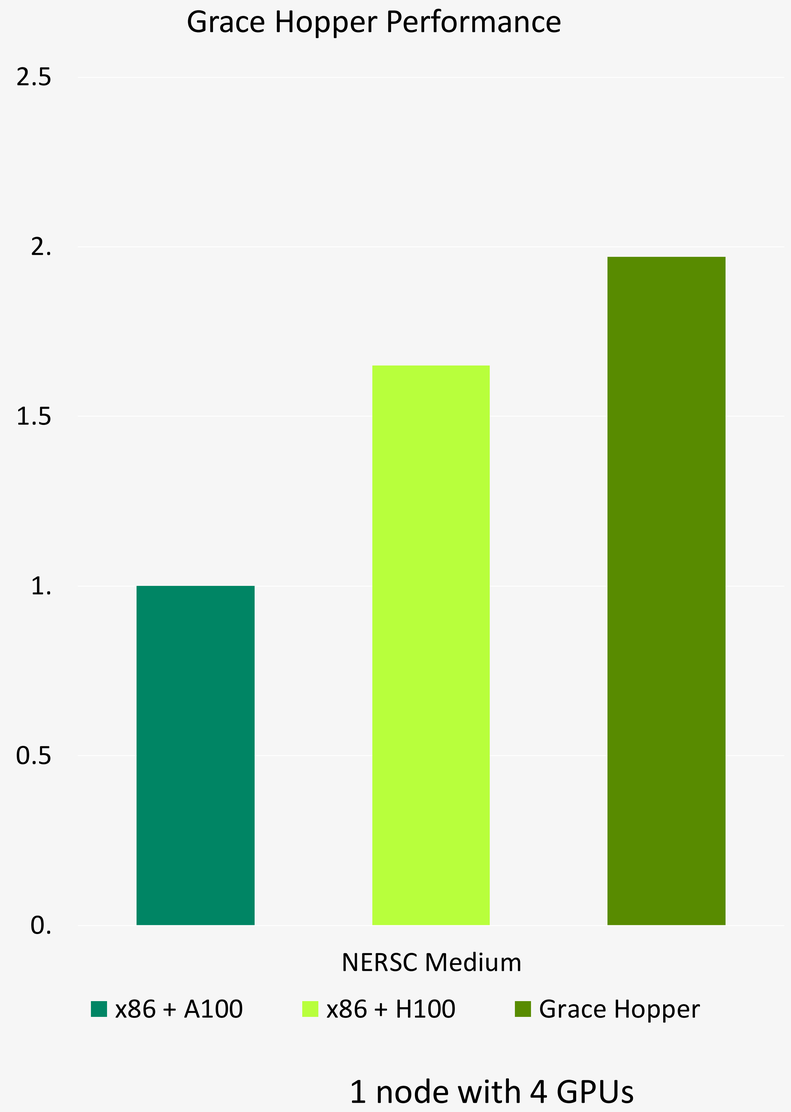


Franz Gross et al., Eur.Phys.J.C 83 (2023), 1125
T. Aoyama et al.Phys.Rept. 887 (2020), 1-166
Fermilab Lattice, HPQCD, and MILC Collaborations, Bazavov et al, Phys.Rev.D 107 (2023) 11, 114514

Lattice QCD with MILC

Fully accelerated using QUDA

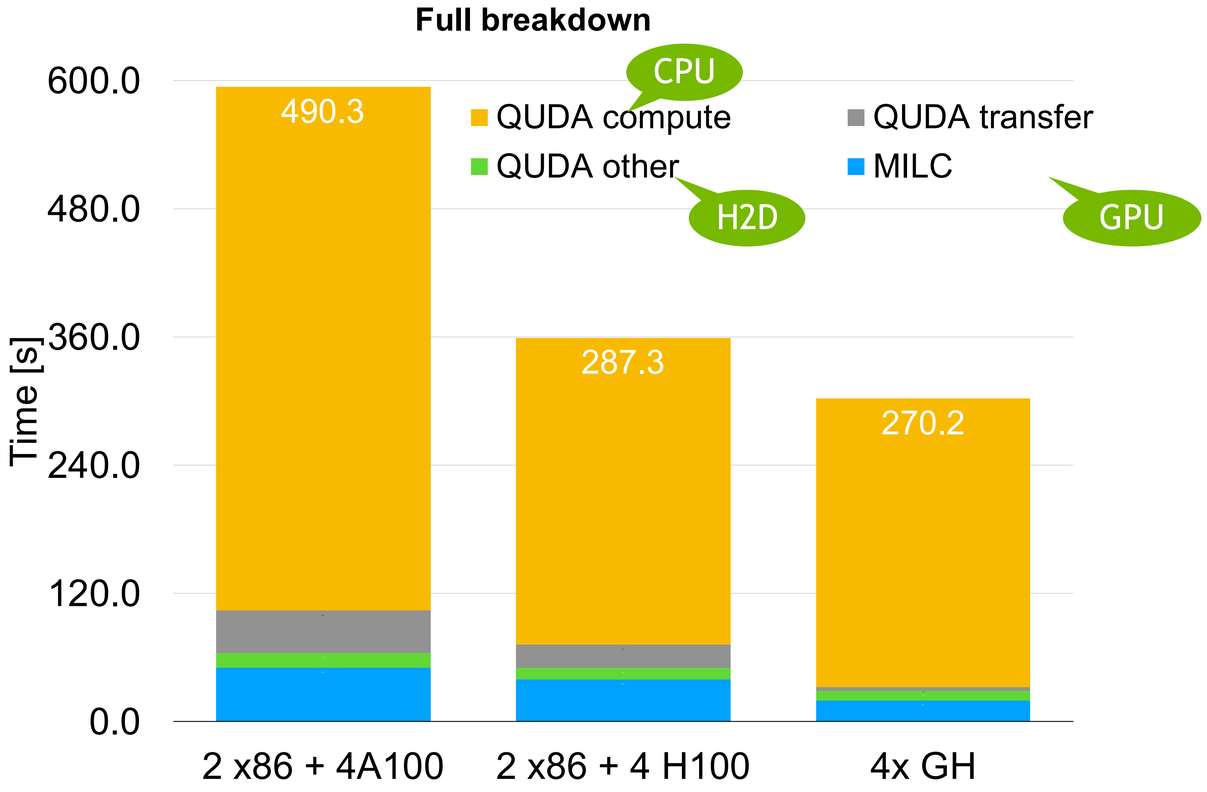
- GPU offload acceleration through QUDA with partial GPU data residency
- Multiple steps in workflow
  - **Generate so-called gauge configurations (requires strong-scaling as generated in a Markov chain)**
  - Calculate physical properties using these configurations as input (throughput problem)
  - Analysis
- **NERSC Medium benchmark (proxy, realistic scale O(100-1000) GPUs)**
- Performance on Grace-Hopper ensures 2x scaling over x86 +A100
  - C2C drastically reduces data-transfer time
  - Grace CPU memory bandwidth accelerates remaining CPU parts
  - Both combined restore scaling between generations

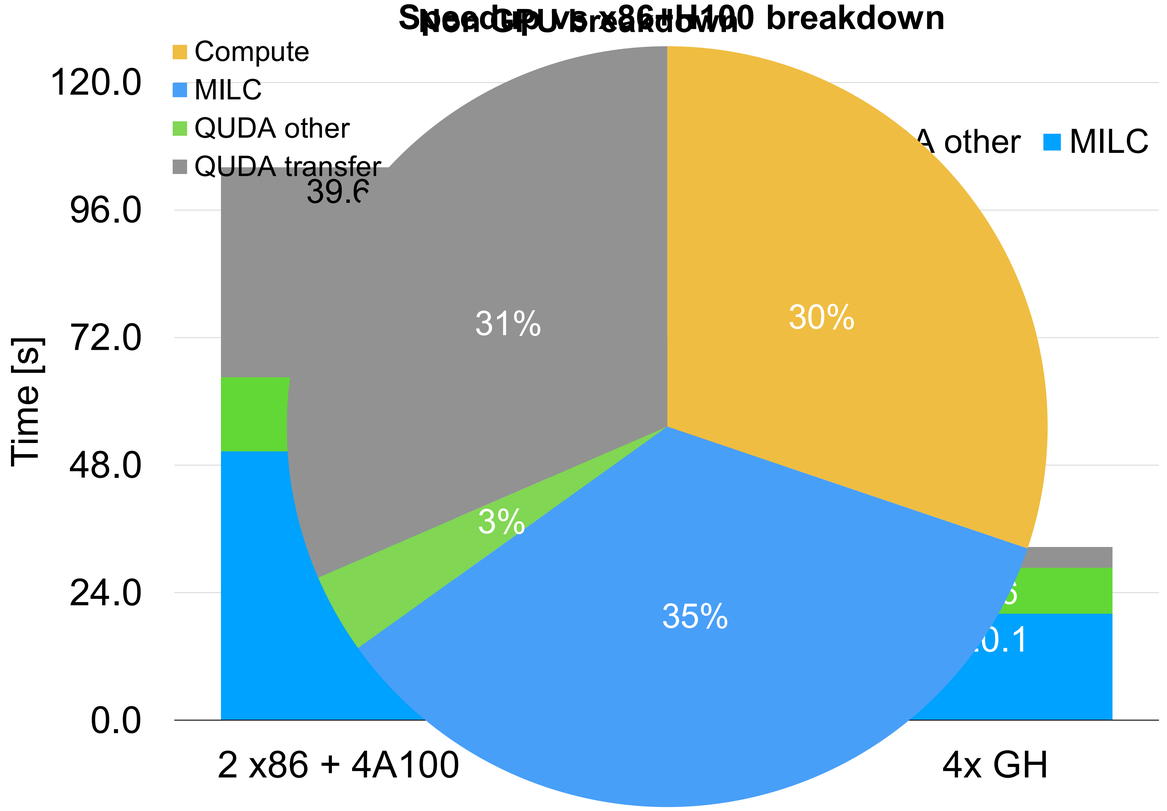


A100 runs were done using AMD EPYC (Rome) CPUs.
H100 runs were done using Intel Xeon (SPR) CPUs.

MILC

Breakdown





A100 runs were done using AMD EPYC (Rome) CPUs.
H100 runs were done using Intel Xeon (SPR) CPUs.

Protein database search

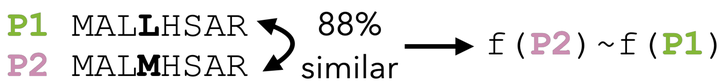
Protein homology inference

Problem definition

We would like to find proteins that perform interesting functions, like digest plastic or bind cancer causing proteins.

The assumption of sequence homology

We have some confidence that two proteins that share sequence similarity will also have similar functions.



The challenge

Assign function to newly sequenced proteins, based on functions of existing protein sequences

Search Top-k similar proteins in a DB

HPC Problem (highly computational expensive)

Optimize Apps

Field Applications

Protein Database Search:

- ADEPT
- BLASTP
- PHMMER

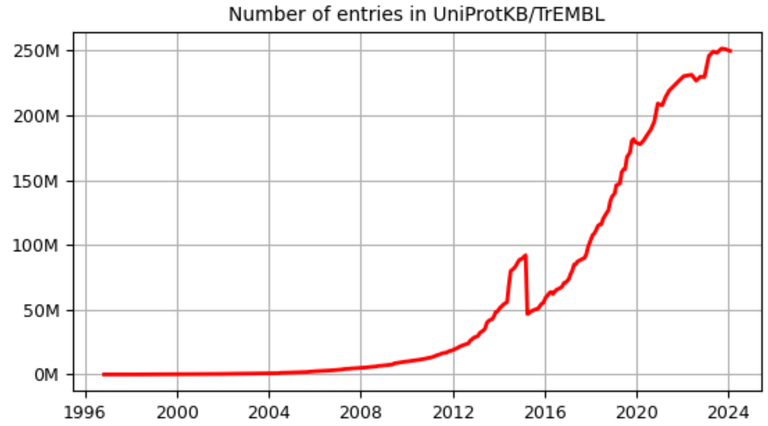
Multi Sequence Alignment:

- MMseqs2
- JACKHMMER

CUDASW4 & LIBMARV

(GPU accelerated)

Requirements to annotate proteins growing:



Accelerate Protein DB Search on GPU

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

NVIDIA

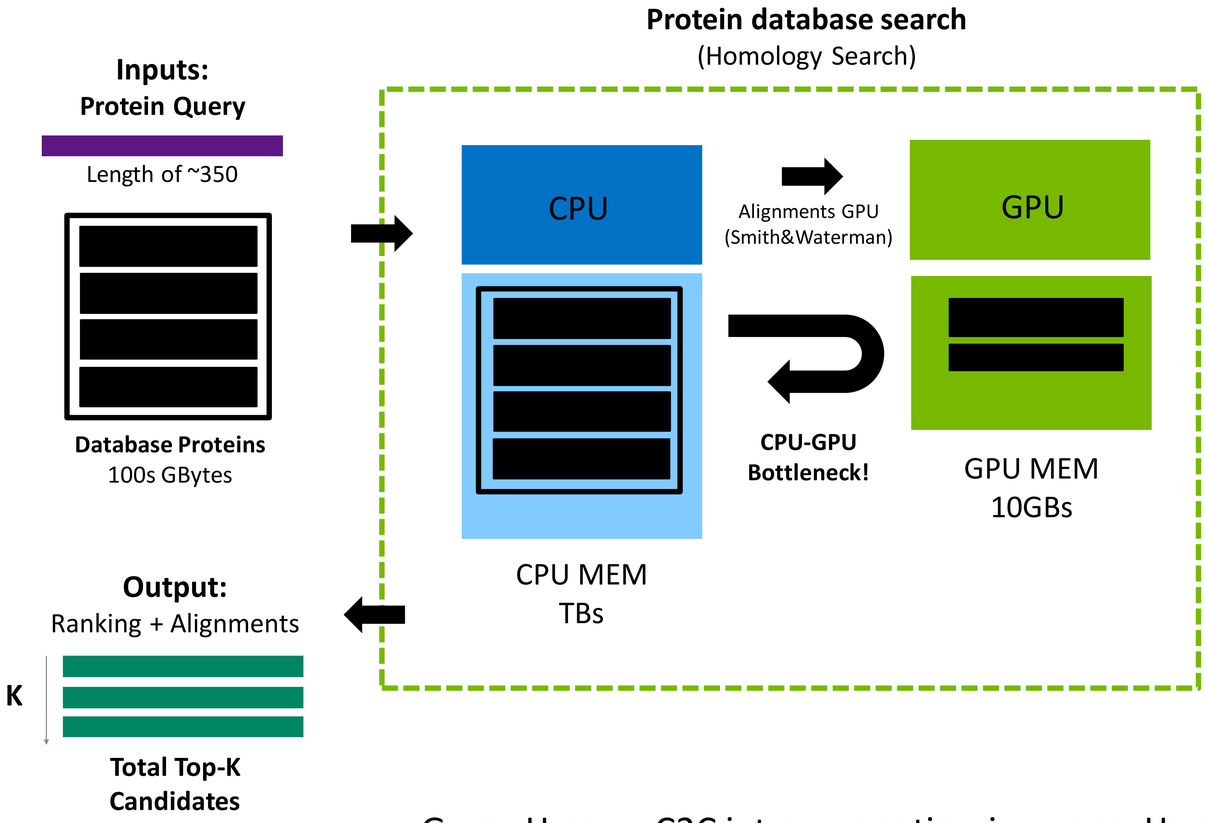
Research collaboration:

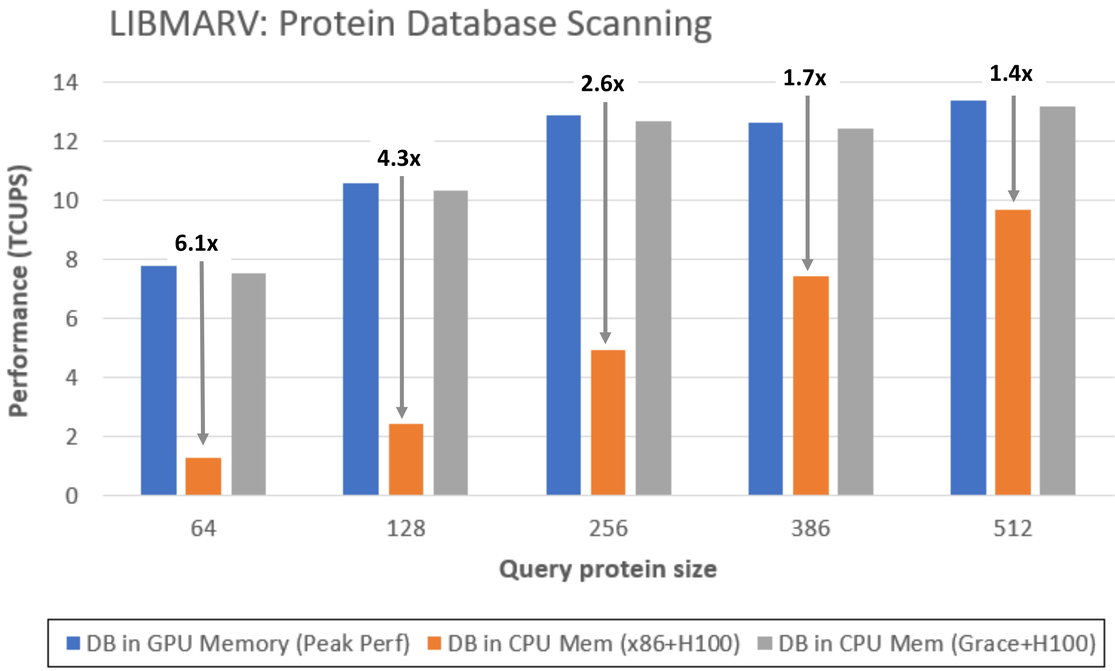
Seoul University, Johannes Gutenberg University, NVIDIA

- **CUDASW++4.0: Ultra-fast GPU-based Smith-Waterman Protein Sequence Database Search**
  Bertil Schmidt, Felix Kallenborn, Alejandro Chacon, Christian Hundt
- **MMseqs2: Sensitive protein sequence searching for the analysis of massive data sets**
  Martin Steinegger, Johannes Söding
- **ColabFold: making protein folding accessible to all**
  Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, Martin Steinegger

Protein database search using CUDASW4 & LIBMARV

Grace-Hopper: 1.4 – 6.1x speedup



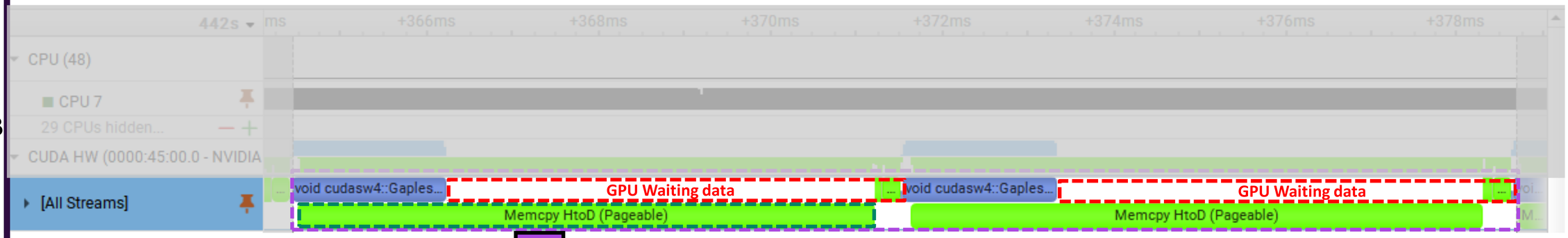


- Grace-Hopper C2C interconnection improves Host-Device ~8x bandwidth compared to PCIe.
- Grace-Hopper achieves peak performance independent of DB size.
- Traditional PCIe interconnection penalize overall performance 1.4 to 6.1 times.
- H100 has DPX support acceleration for these applications.

Protein database search

Performance Profile – Protein Query: 128 Size

x86 + H100 (H100 80GB HBM3)

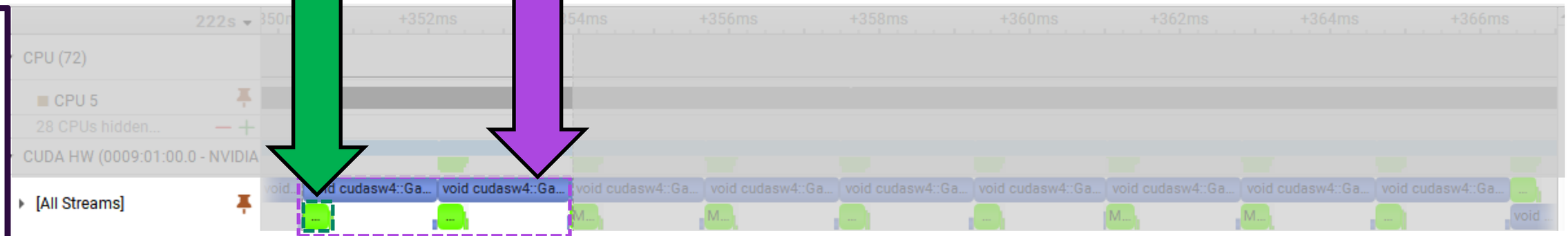


CPU-GPU Transfers
17x faster

19 GB/s

DB processing
4.2x faster

Grace Hopper (H100 96GB HBM3)

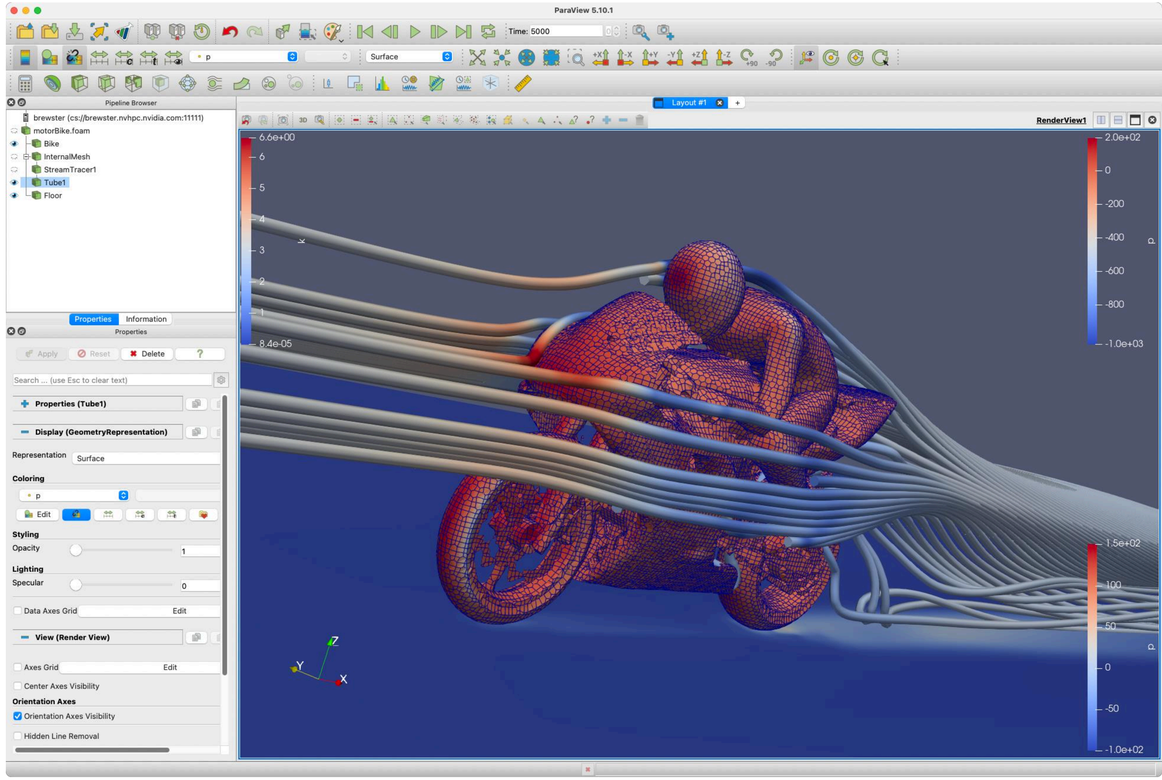


352 GB/s

Computational Fluid Dynamics

Openfoam

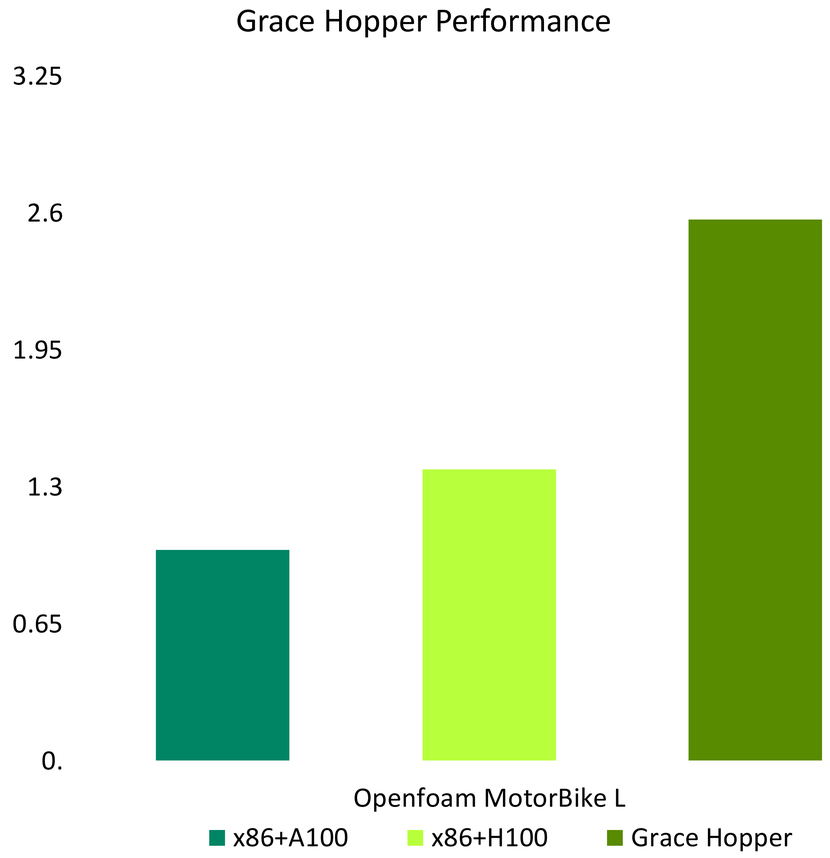
- Computational fluid dynamics (CFD) toolbox
  - Developed by ESI-OpenCFD
- Popular in automotive and other engineering sectors
- Highly configurable fluid flow solvers with turbulence / heat transfer / etc.
- Can leverage GPU-accelerated AmgX linear solvers via plugin interface (PETSc4FOAM)



OpenFoam

Partially GPU Accelerated – mostly CPU limited

- HPC motorbike problem (Large)
  - Solves with the simpleFoam application
  - Around 30-40% of CPU-only execution is spent in linear solves
  - Hybrid approach spends large proportion of time on the CPU
- Performance on Grace Hopper
  - High CPU and GPU memory bandwidth improve compute performance
  - C2C bandwidth minimizes the cost of migrating CPU matrix data



~35M cells benchmark designed by OpenFOAM HPC technical committee

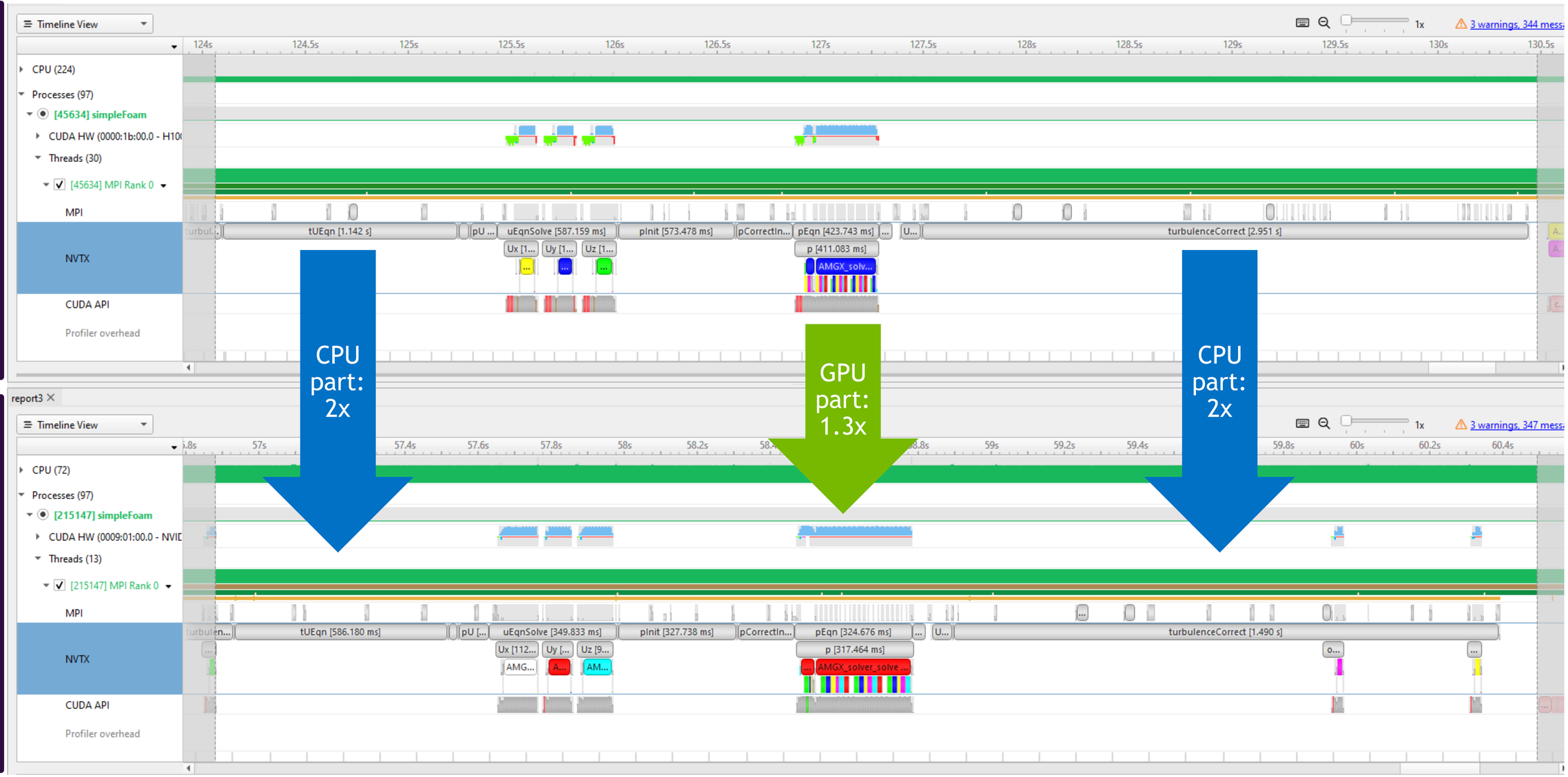
A100 runs were done using AMD EPYC (Rome) CPUs.
H100 runs were done using Intel Xeon (SPR) CPUs.

OpenFoam

Nsight Systems Profile

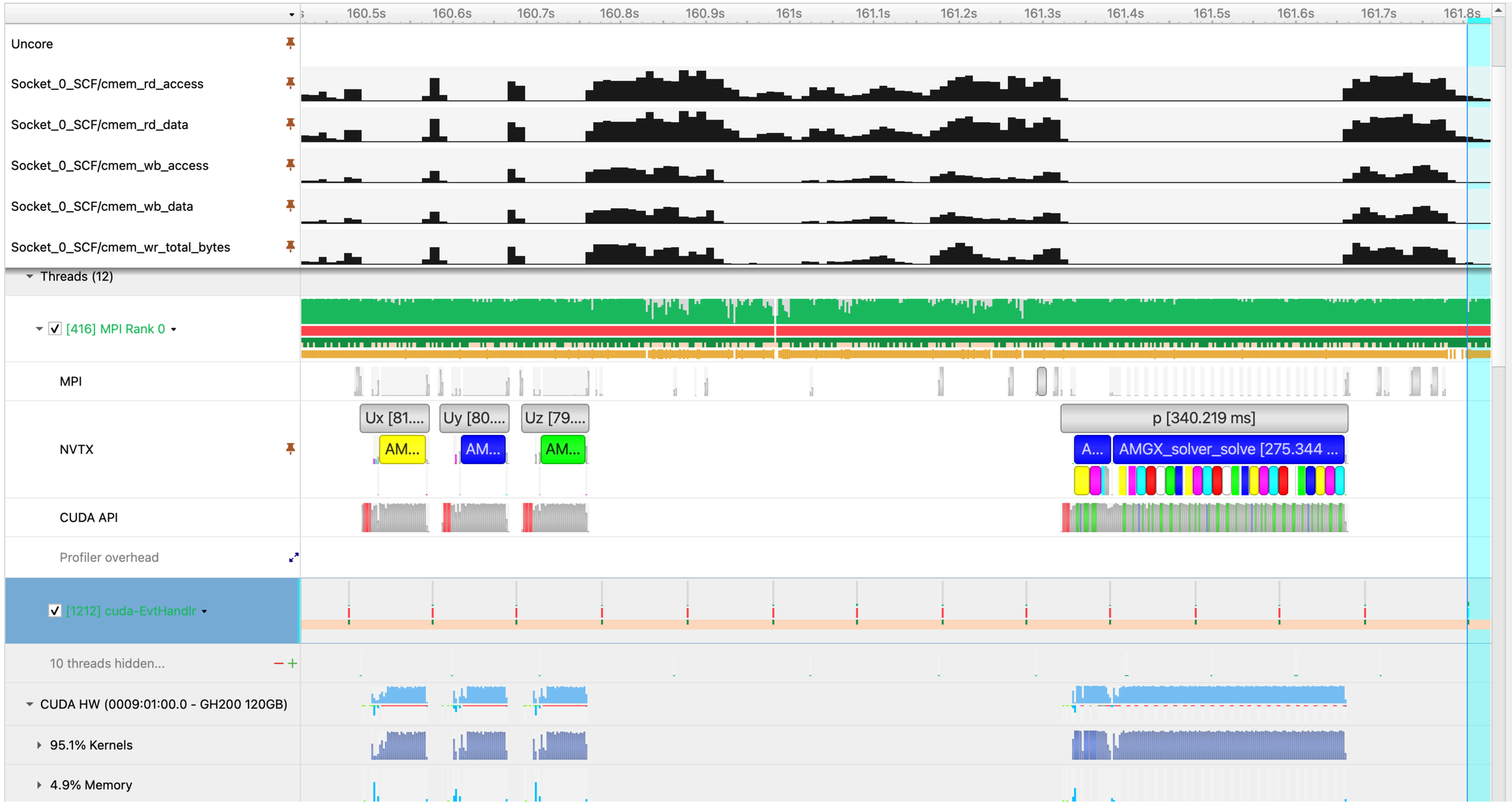
x86 + H100 (H100 80GB HBM3)

Grace Hopper (H100 96GB HBM3)



OpenFoam

Nsight Systems overview



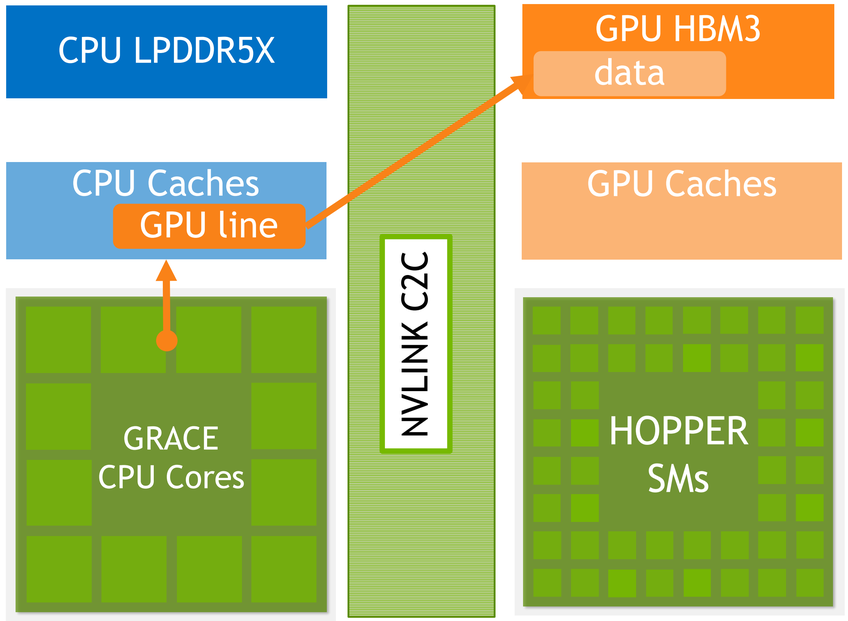
Application on Accelerated Systems

Coherently GPU Accelerated

- Exploit GPU / CPU coherency
- Use all available system features
  - not necessarily clean distinction between phases

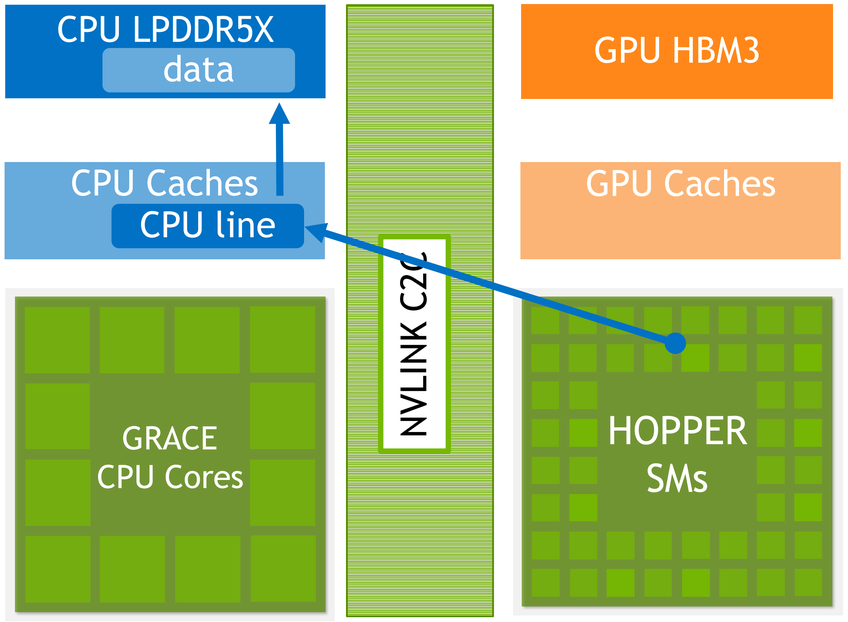
Global Access to All Data

Cache-coherent access via NVLink C2C from either processor to either physical memory



Grace directly reading Hopper's memory

CPU fetches GPU data into CPU L3 cache
Cache remains coherent with GPU memory
Changes to GPU memory evict cache line

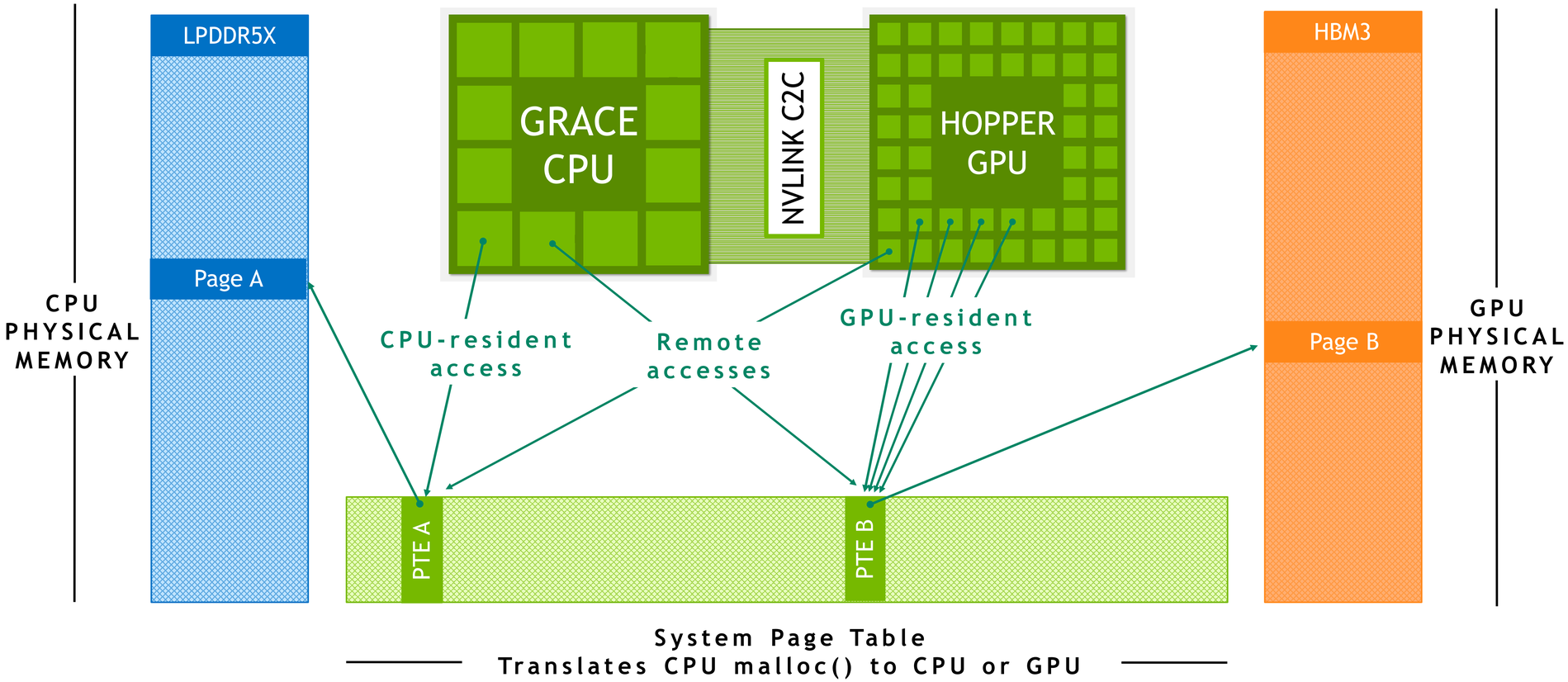


Hopper directly reading Grace's memory

GPU loads CPU data via CPU L3 cache
CPU and GPU can both hit on cached data
Changes to CPU memory update cache line

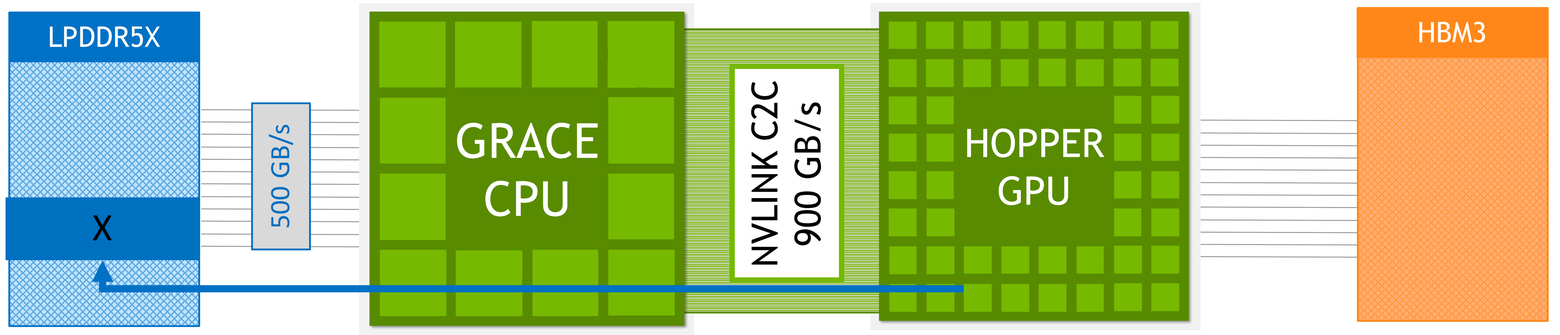
Grace Hopper

Address Translation Service (ATS) enables full access to all CPU & GPU allocations
Migrations are not required: Fewer Migrations

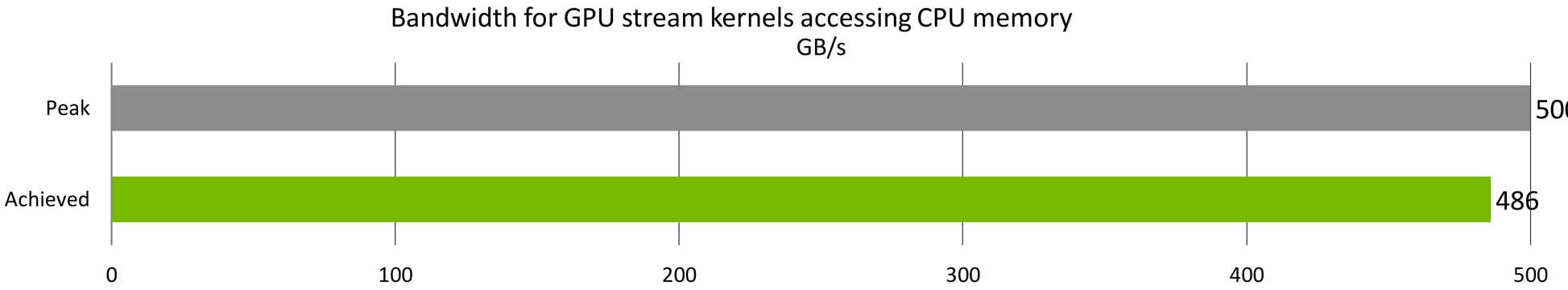


ATS creates a single page table for the whole system
NVLink C2C allows access to all physical memory without migration

High Bandwidth Memory Access & Automatic Data Migration

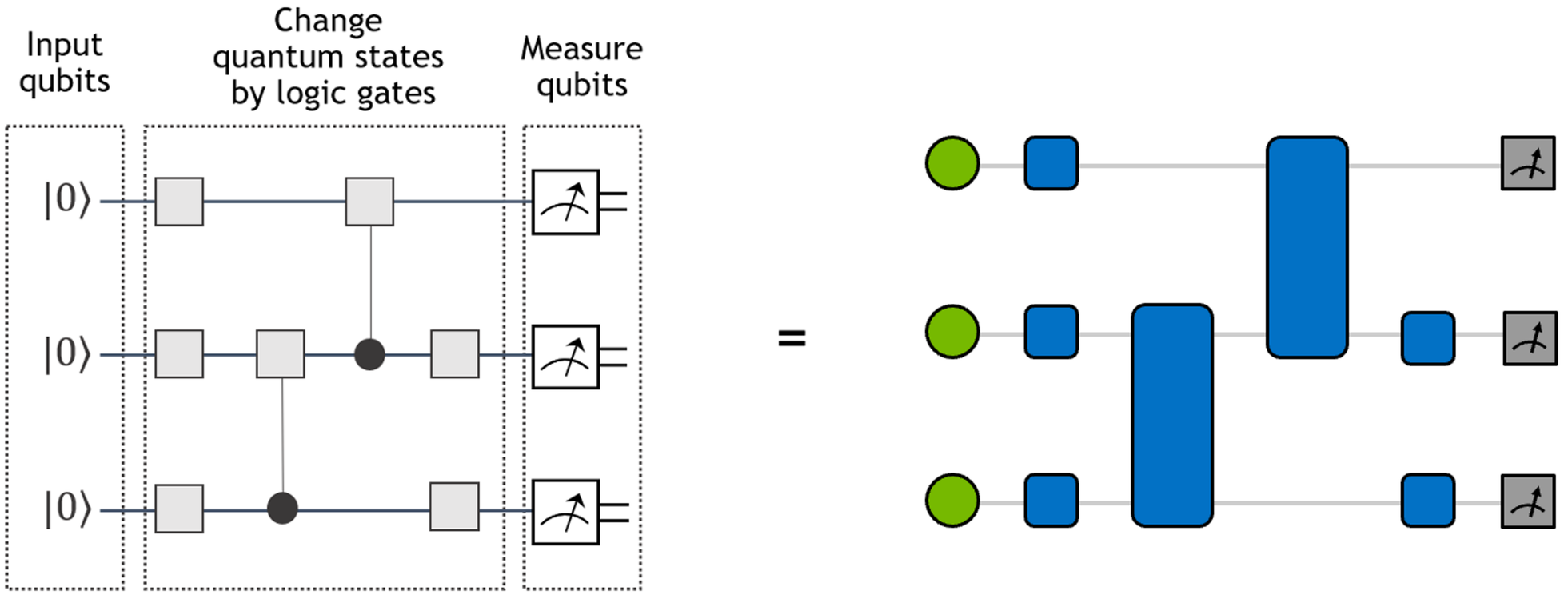


Hopper can access Grace memory
at full CPU memory speed of 500 GB/sec



Quantum Computing

Tensor Network-Based Circuit Simulation



- Quantum Circuit Simulation: Represent qubits, quantum gates, and measurements using tensors.
- Tensor Network Contractions: Execute sequential pairwise tensor contractions following a predetermined order.
- Tensor Network Index Slicing: Decrease memory requirements while incurring extra computational overhead.

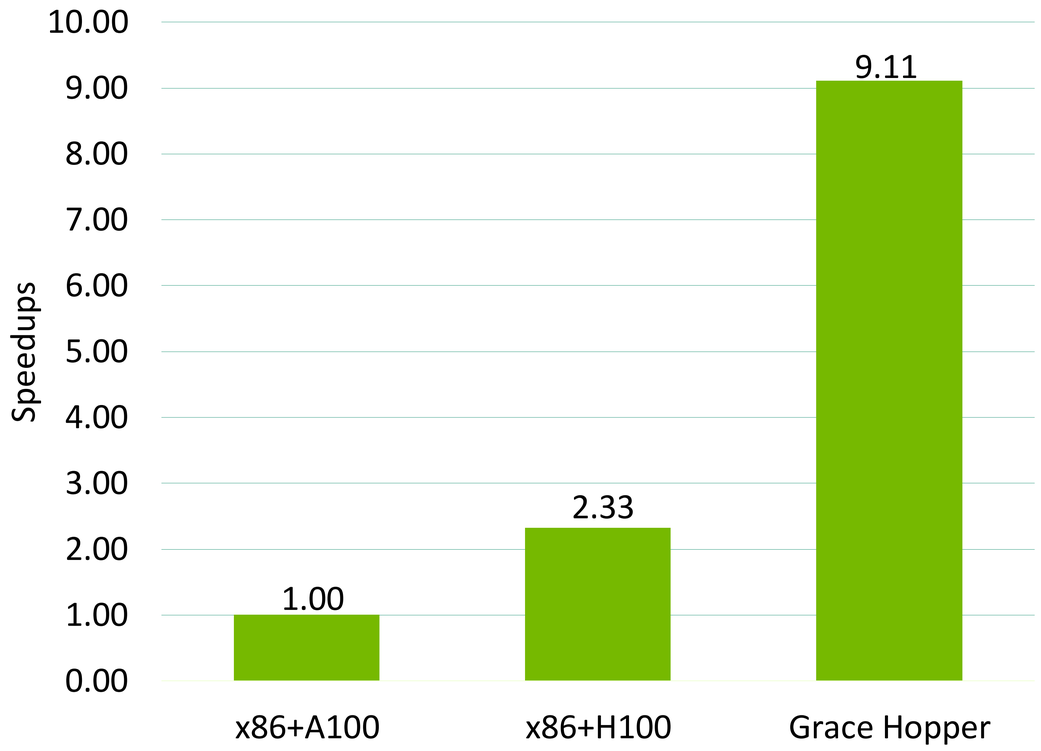
Overview — NVIDIA cuQuantum 23.03.0 documentation

Quantum Computing

Example: Sycamore 53-qubit 20-cycle circuit

	x86+A100/H100	Grace Hopper
Tensor locations	Device	Host + Device
Largest tensor size	32GB	256GB
Overall computing complexity	1.00	0.14

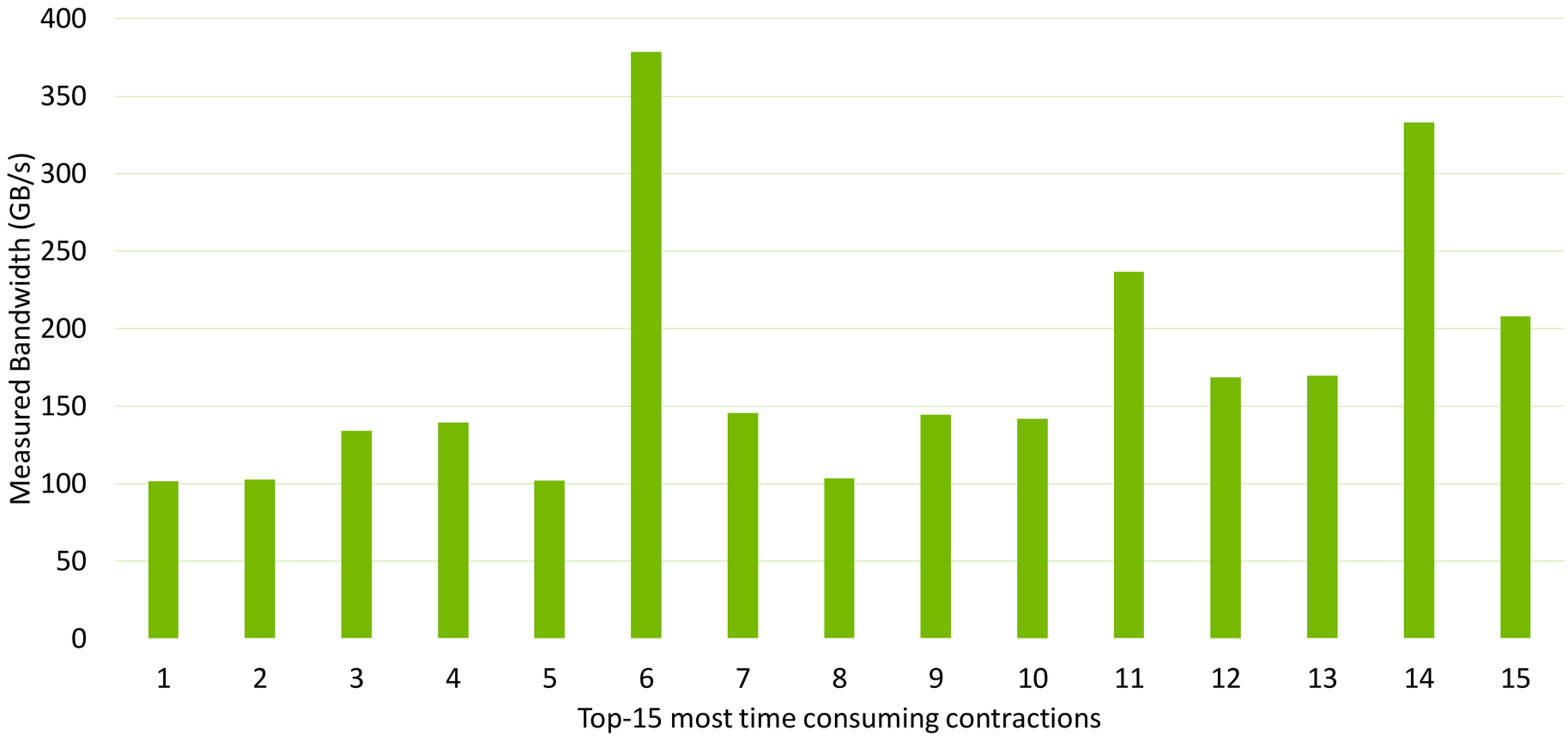
- C2C's high bandwidth enables rapid access to host memory, integrating host and device memory into a unified system on Grace Hopper without migration.
- The larger memory pool simplifies the computation for tensor network contraction order, overcoming device memory size constraints.
- On Grace Hopper, a 256GB configuration demonstrated a 3.92x speedup compared to a 32GB configuration on an x86+H100 system.



Quantum Computing

Example: Sycamore 53-qubit 20-cycle circuit

- Measured C2C bandwidth in top-15 contractions



Backfill free CPU resources

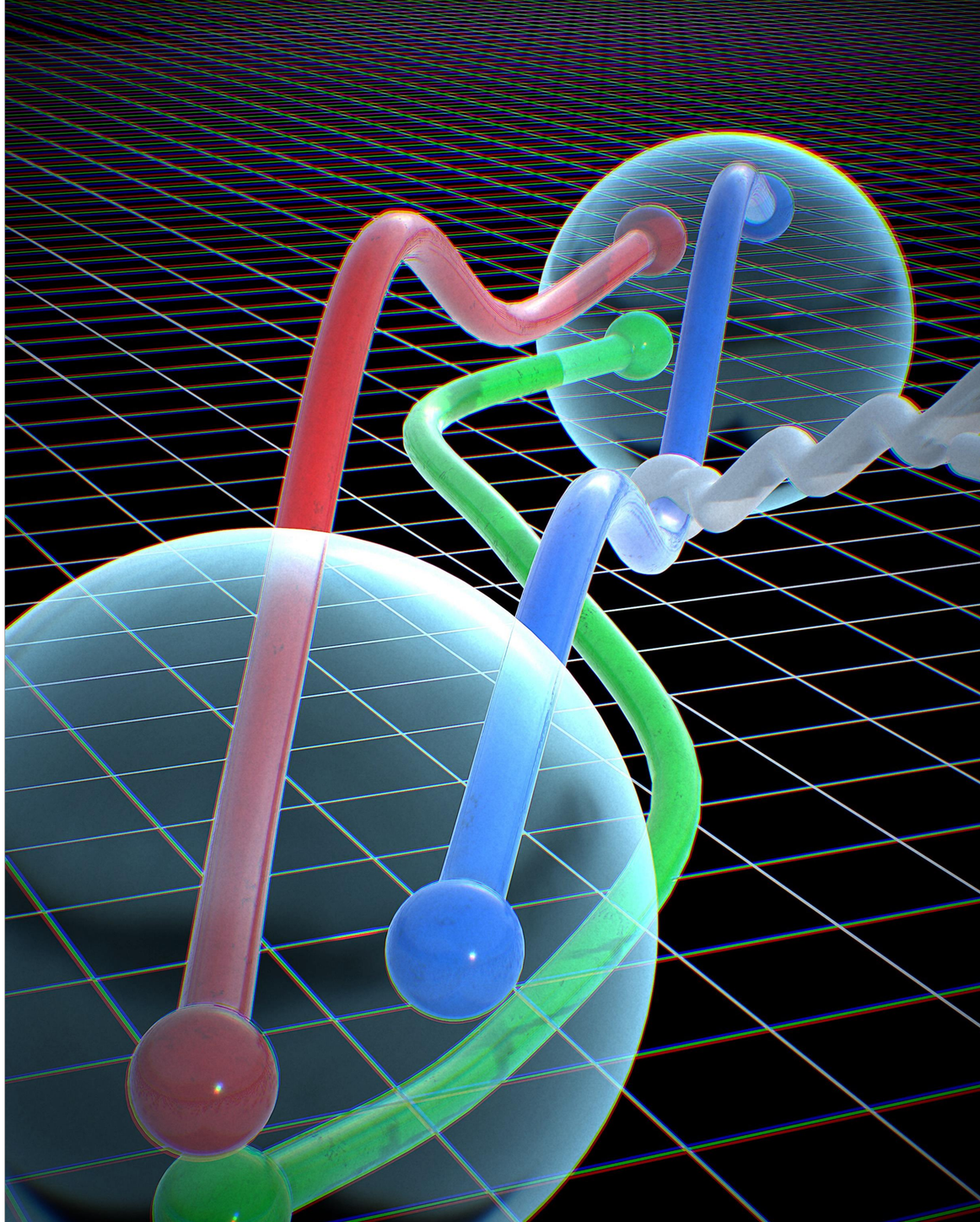
Run highly demanding phase on GPU and overlap another with another phase on the CPU

- Accelerated jobs mostly uses the GPU and only fraction of CPU

- Backfill idle CPU resources

Neutron Decay

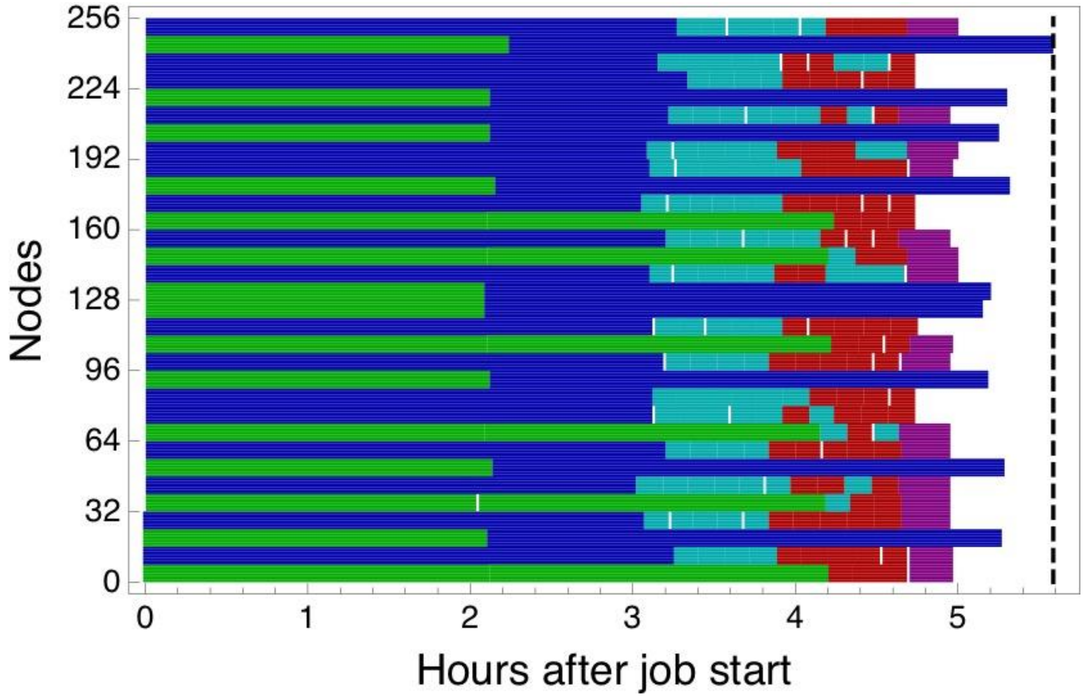
- Neutrons decay into protons, an electron and an anti-neutrino. Mean time ~881 seconds.
- The exact value is an important input in understanding the evalution of the early universe
- Initially we only had H (1p) and He (2p2n) atoms
  - Why that particular ratio
- Understand experimental measurements and theoretical (standard model) predictions
- An important input for this can be theoretically determinded using Lattice QCD
- High precision required to identify deviations between experiment and theory



METAQ and MPIJM

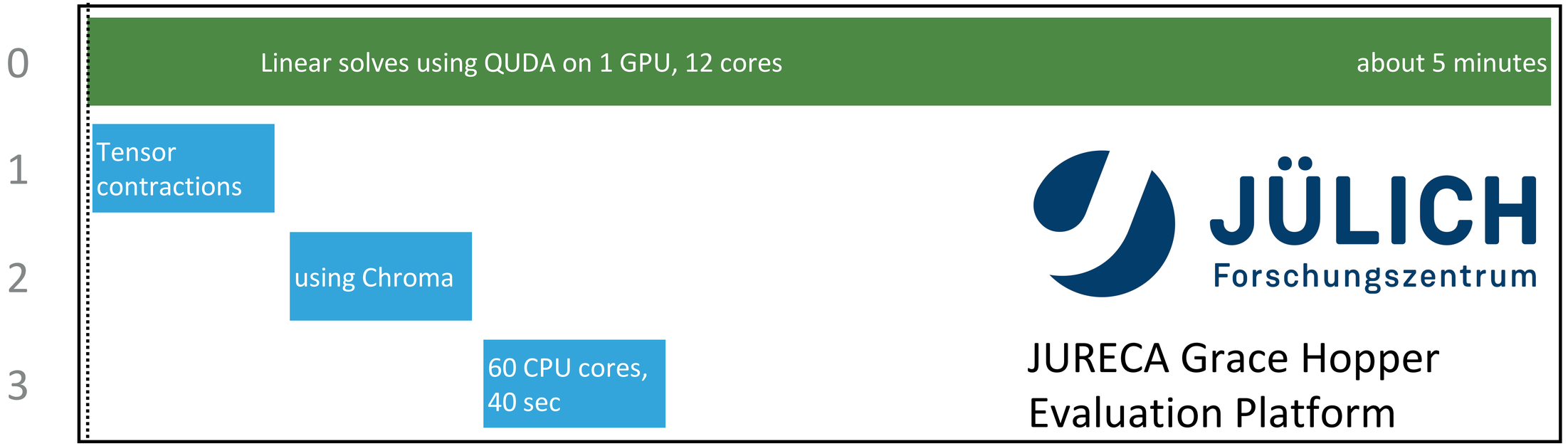
https://callat-qcd.github.io/software/software/

- need to run hundreds of thousands to millions of independent, small node tasks
- “If we stack GPU tasks after CPU and so on, much of the wall-clock time will be wasted in the sense that the GPUs will not be in use - and the CPU wall-clock time is approaching a significant fraction of the GPU wall-clock time”
- “Job bundling wastes significant amounts of wall-clock time as performance of each task can vary substantially if nodes are close together or far apart”
- METAQ and MPI_JM jobmanager to intelligently backfill CPU resources and balance workloads
- Grace Hopper allows to fully hide CPU needs behind needed GPU jobs with limited CPU needs



METAQ

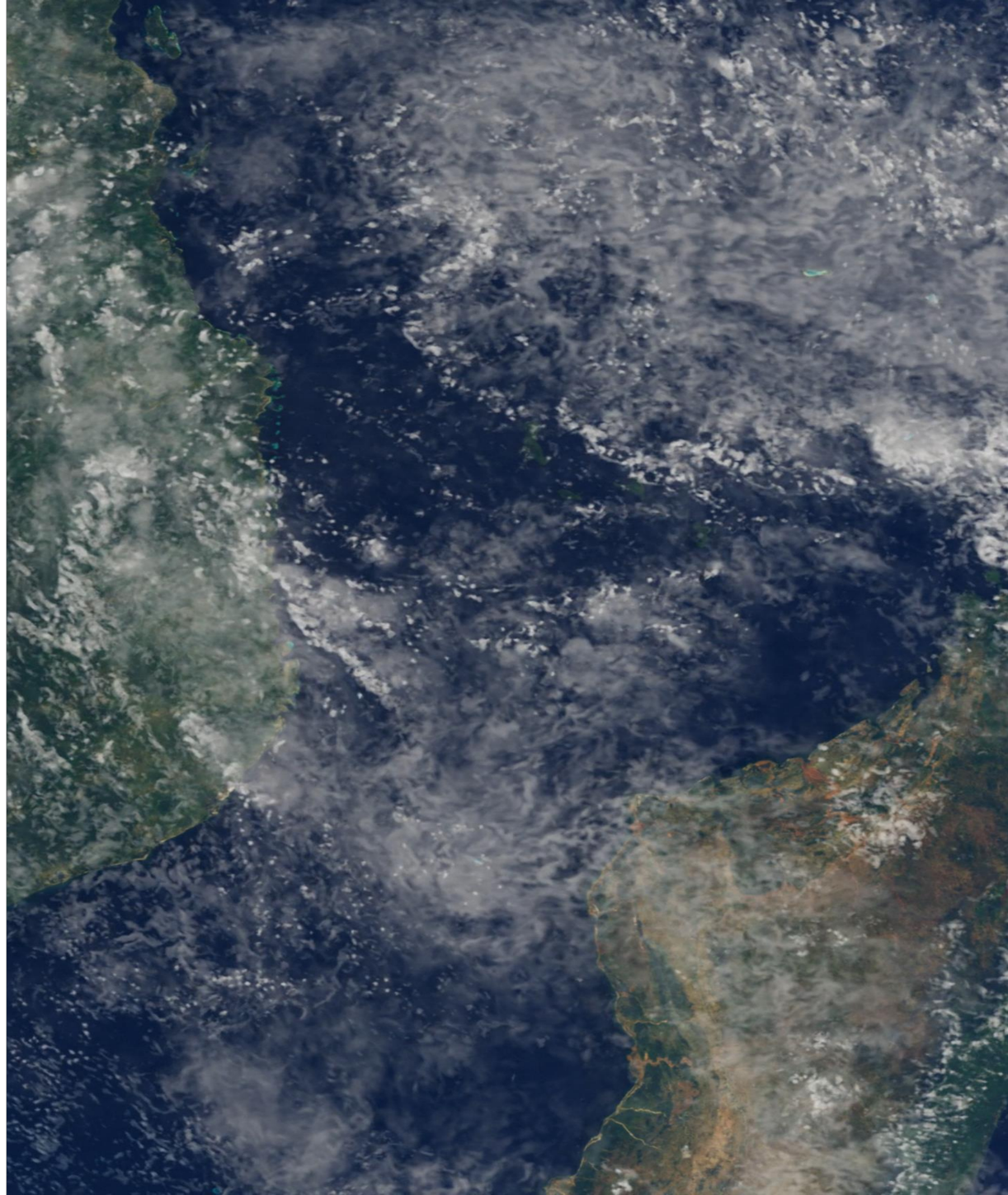
Fully hiding CPU needs



From on LLView report

ICON Coupled Ocean

- ICON is a unified next-generation global numerical weather prediction and climate modelling framework
- Developed by DWD (German weather prediction center), MPI-M (German Max Planck climate research institute) and MeteoSwiss with help from CSCS
- Currently used for operational forecast at DWD, soon to be in production in Switzerland on GPUs. Used by many institutes for climate simulations
- Typical scales from 1 to 1000s GPUs
- Atmospheric simulation is fully GPU-ported with OpenACC. Ocean part is not fully ported yet and can only be run on the CPU
- Coupled atmosphere-ocean simulations are very important for understanding long-term climate change and multiple institutions are currently working on such setups



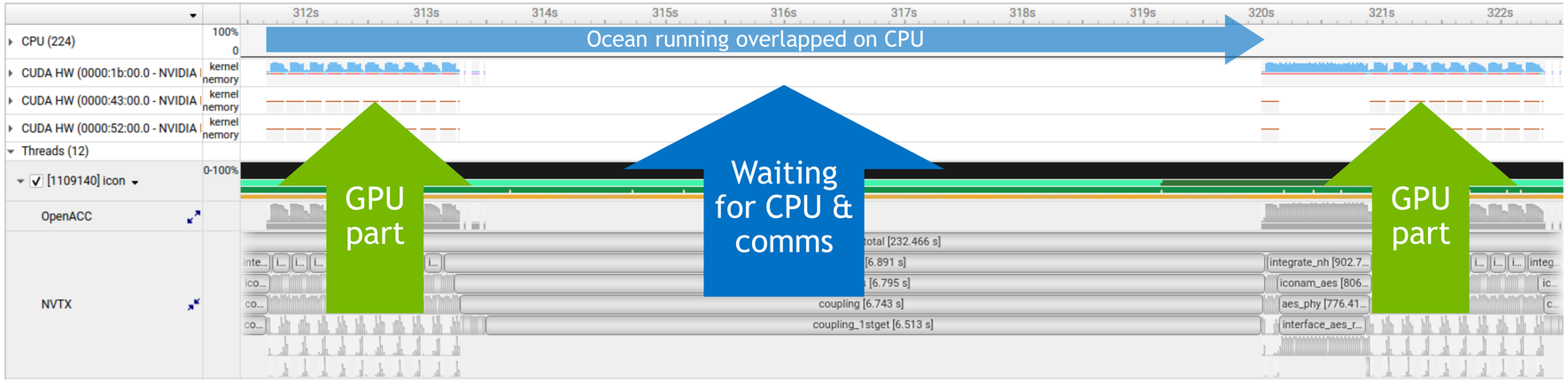
ICON Coupled Ocean

Profile

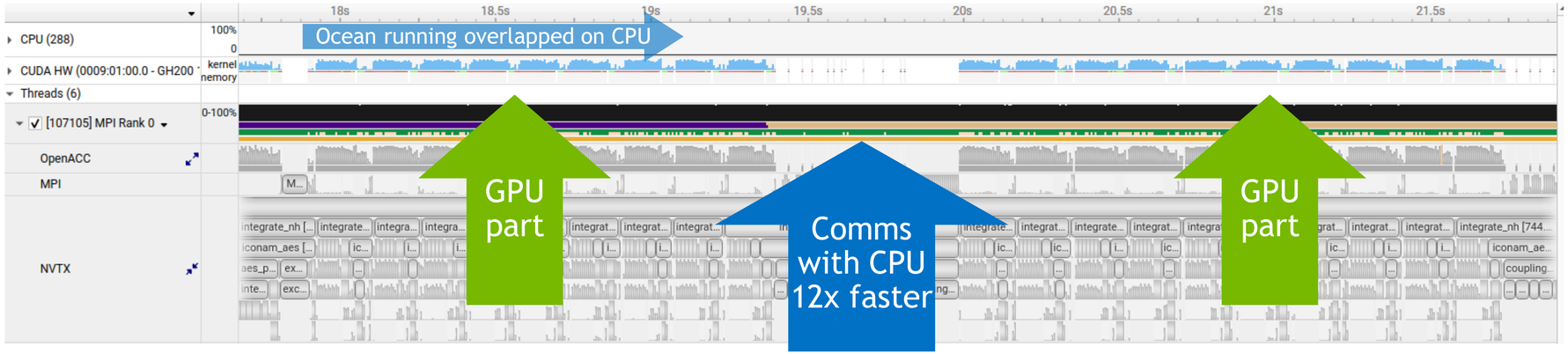
- Full globe coupled simulation at 10 km atmosphere resolution and 5 km ocean resolution. 90 vertical atmosphere layers, 72 vertical ocean layers. Atmosphere time-step is 90s, ocean time-step is 5 min and coupling time-step is 15 min. Atmosphere and ocean run in different ranks within the same MPI job. 64 GPUs and 512 (Eos) or 3008 (Alps) CPU ranks

Profiling atmosphere GPU process

Eos



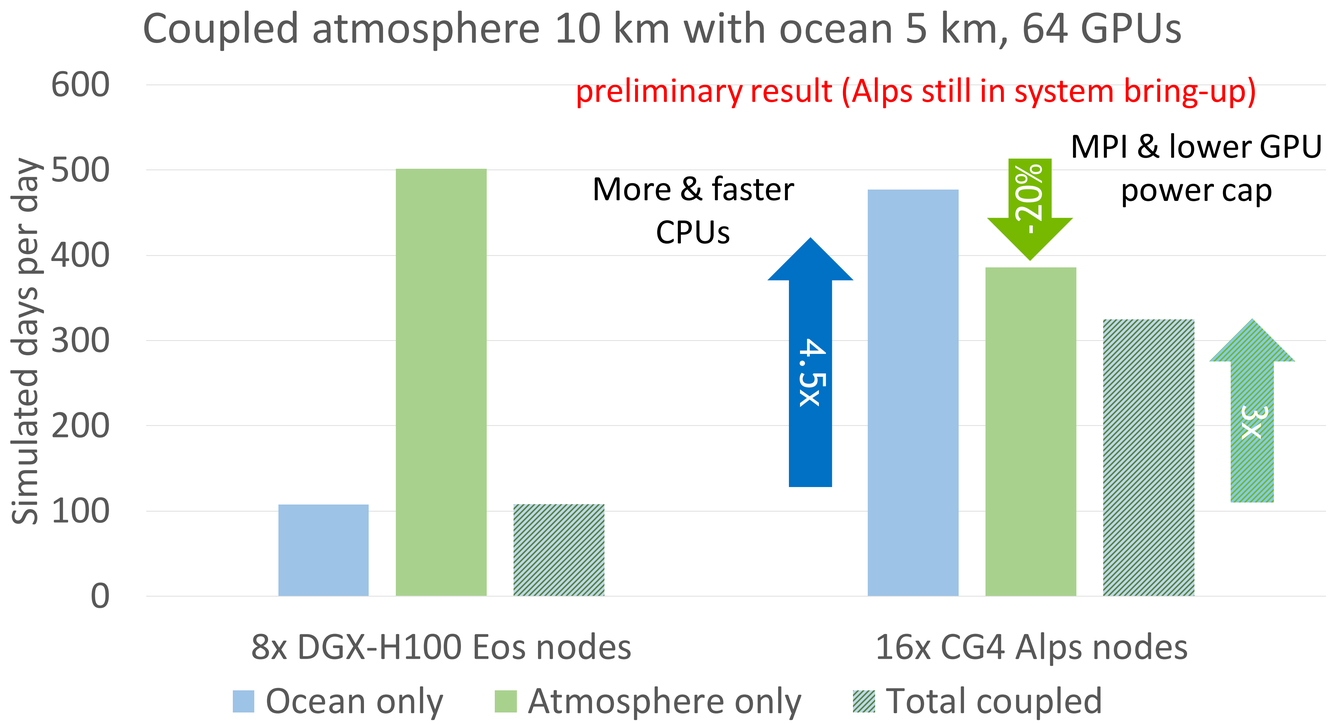
Alps



ICON Coupled Ocean

Grace-Hopper: 3x speedup

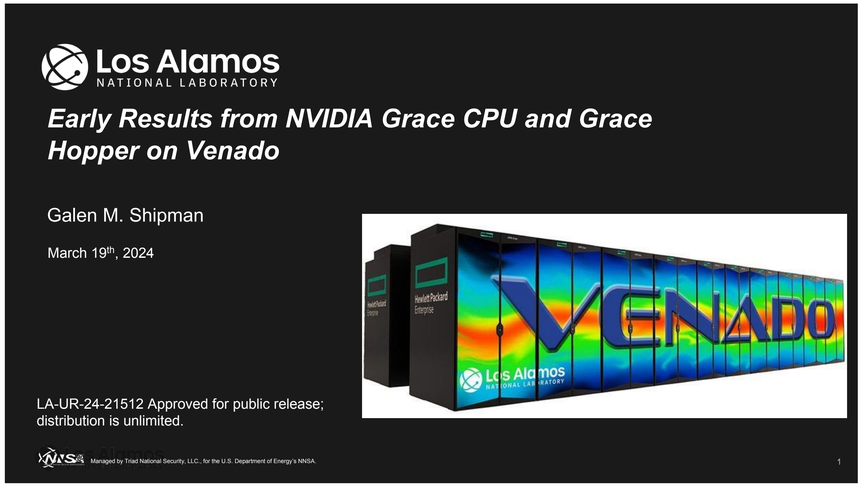
- On EOS:
  - Performance limited by Ocean running on the CPU
- On Alps:
  - Unleash full performance of Hopper GPUs
  - Grace is powerful enough to run the ocean in the background
  - Alps network is still in bring-up phase, which introduces some atmosphere-only and coupling overhead
  - 3x end-to-end performance



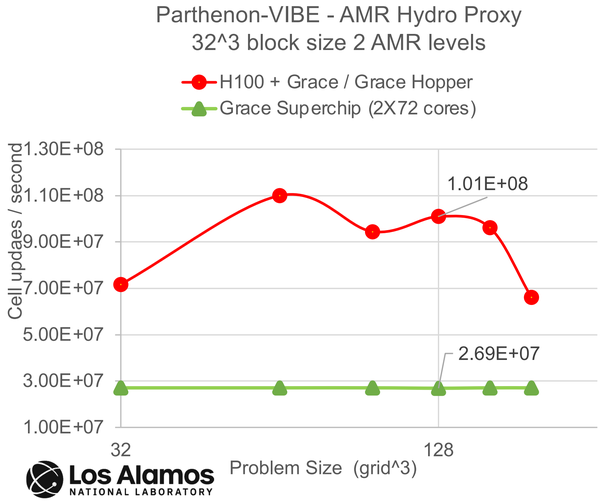
See also: Thomas Schulthess, CSCS – S62157

More science on Grace Hopper at GTC

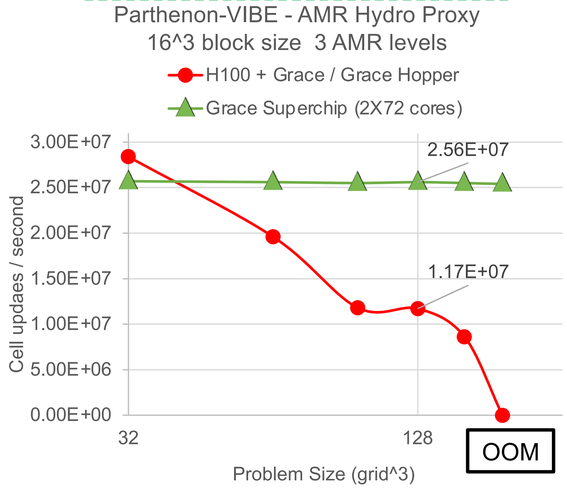
Galen M. Shipman, Los Alamos - S62247



Regular Computation



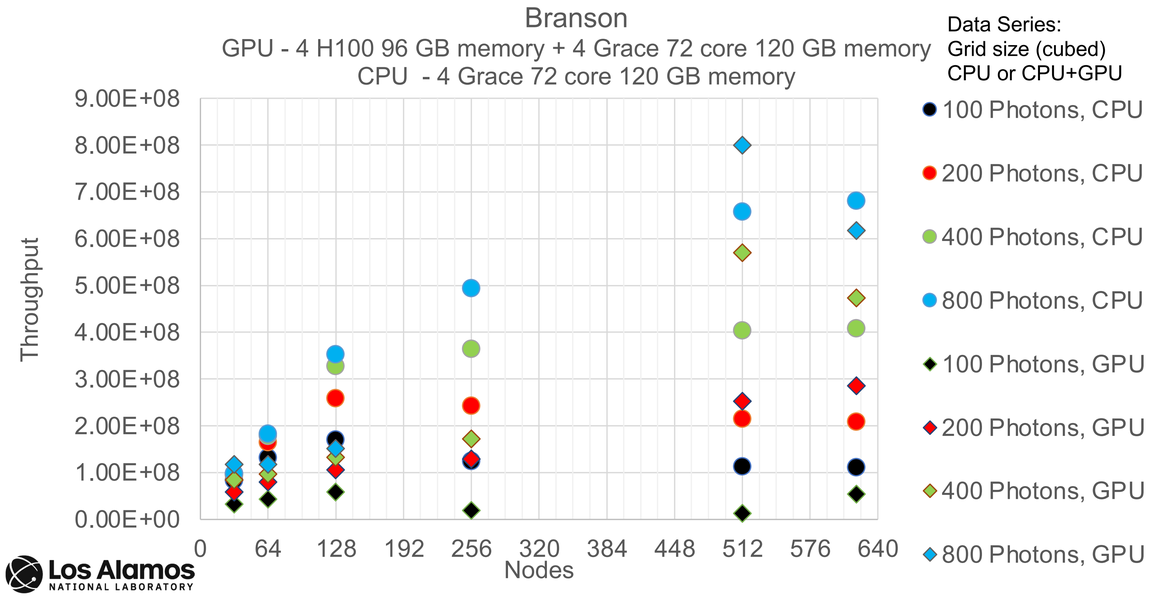
Irregular Computation



Los Alamos NATIONAL LABORATORY

4

Demonstrated Scalability to 620 nodes, 2,480 G/H



Los Alamos NATIONAL LABORATORY

5

NVIDIA

More science on Grace Hopper at GTC

Lars Koesterke, TACC - S61598

TACC
TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

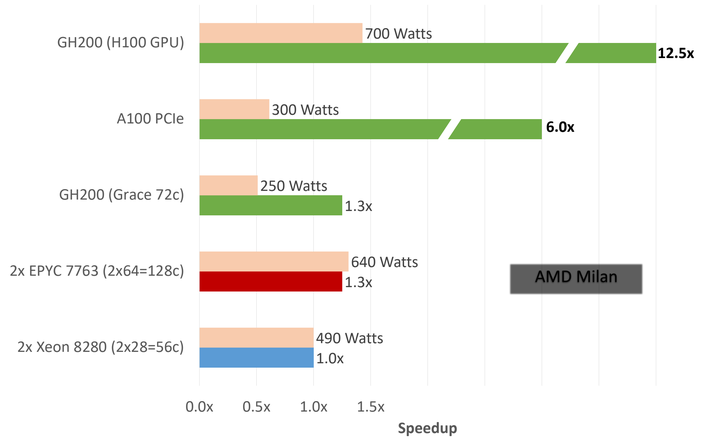
Scientific Computing with NVIDA Grace and Arm Ecosystem

GTC conference
March 2024

Lars Koesterke, w/ Junjie Li, Hanning Chen, John Cazes, and many others
Texas Advanced Computing Center
The University of Texas at Austin

NAMD

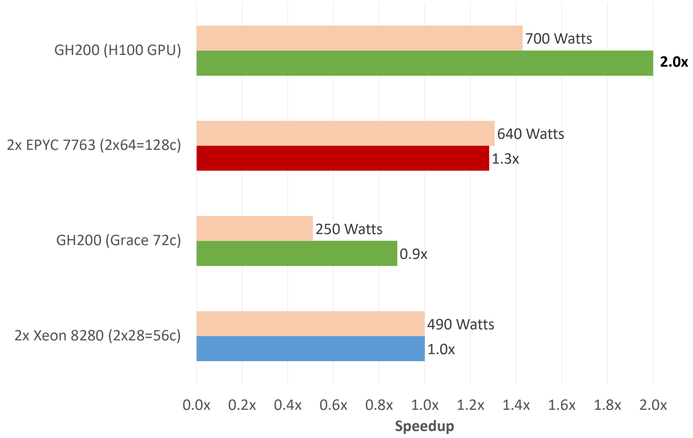
- **Molecular Dynamics**
- NAMD 3.0b3, STMV ~1M atoms
- CPU-only GH200 (72 cores) is comparable to dual EPYC 7763 (128 cores)



Grace offers very good energy efficiency compared to other CPUs.

SeiSol

- **Computational seismology**
- CPU optimized code getting ported to GPU.
- Heavily relies on small matrix multiplication (libxsmm).
- Opportunity for small matrix multiplication optimization in NVPL



Grace offers comparable performance with less energy consumption

NVIDIA

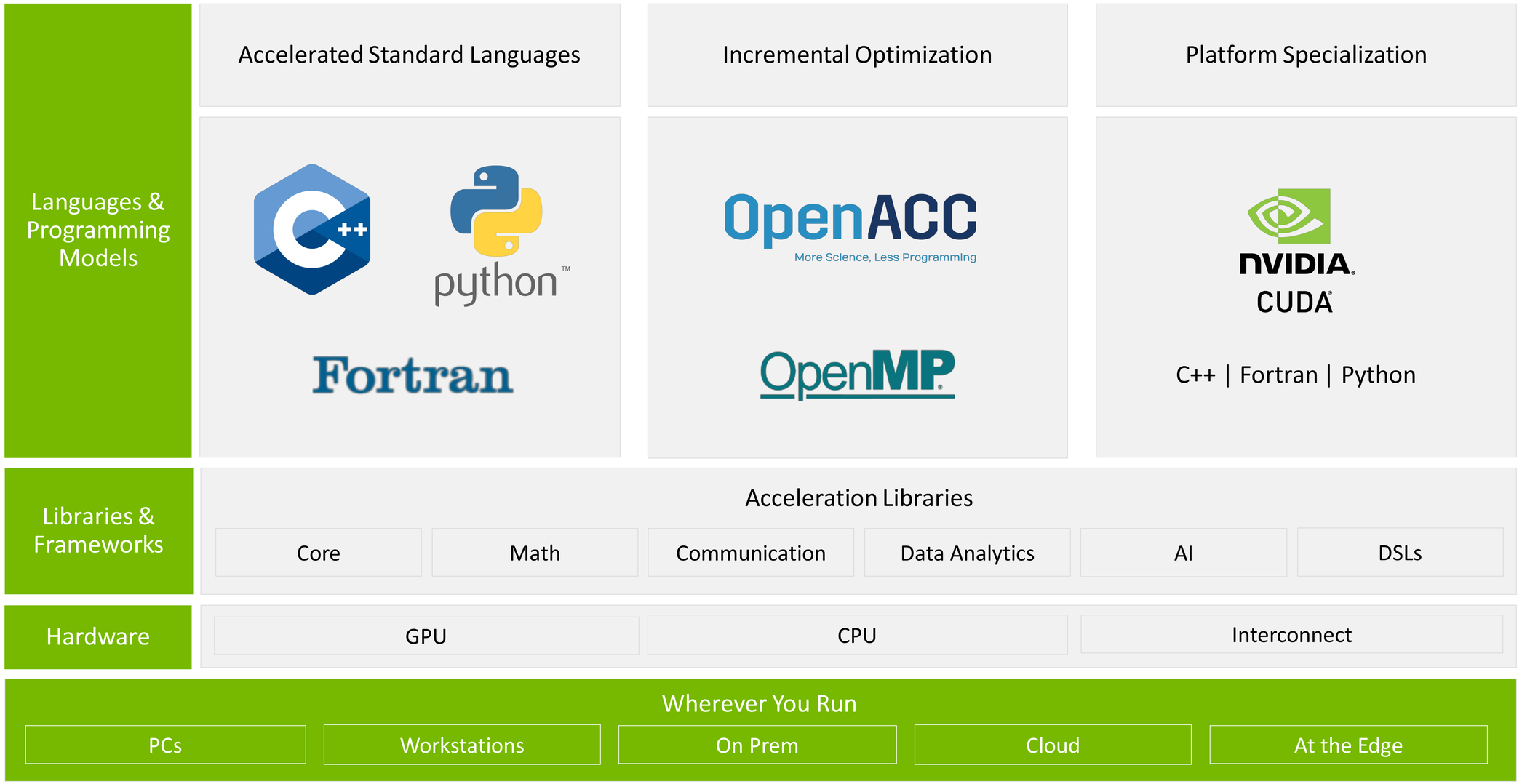
Getting ready for Grace-Hopper

Recompile and Run

- Currently existing applications do not need to be changed
  - Recompile the application for ARM Neoverse-V2 (Grace) and sm_90 (Hopper)*
  - Benefit from more bandwidth everywhere

- Accelerate existing applications
  - Easier to port than ever
  - Large selection of programming models and language available
  - Hardware coherency
  - Obtain overall speedup even for partially ported applications with the Grace CPU and C2C
  - Large selection of tools (NVIDIA tools and 3rd party) available
  - Balanced architecture results in fewer Amdahl’s limiters

Programming the NVIDIA Platform



Choosing A Programming Model

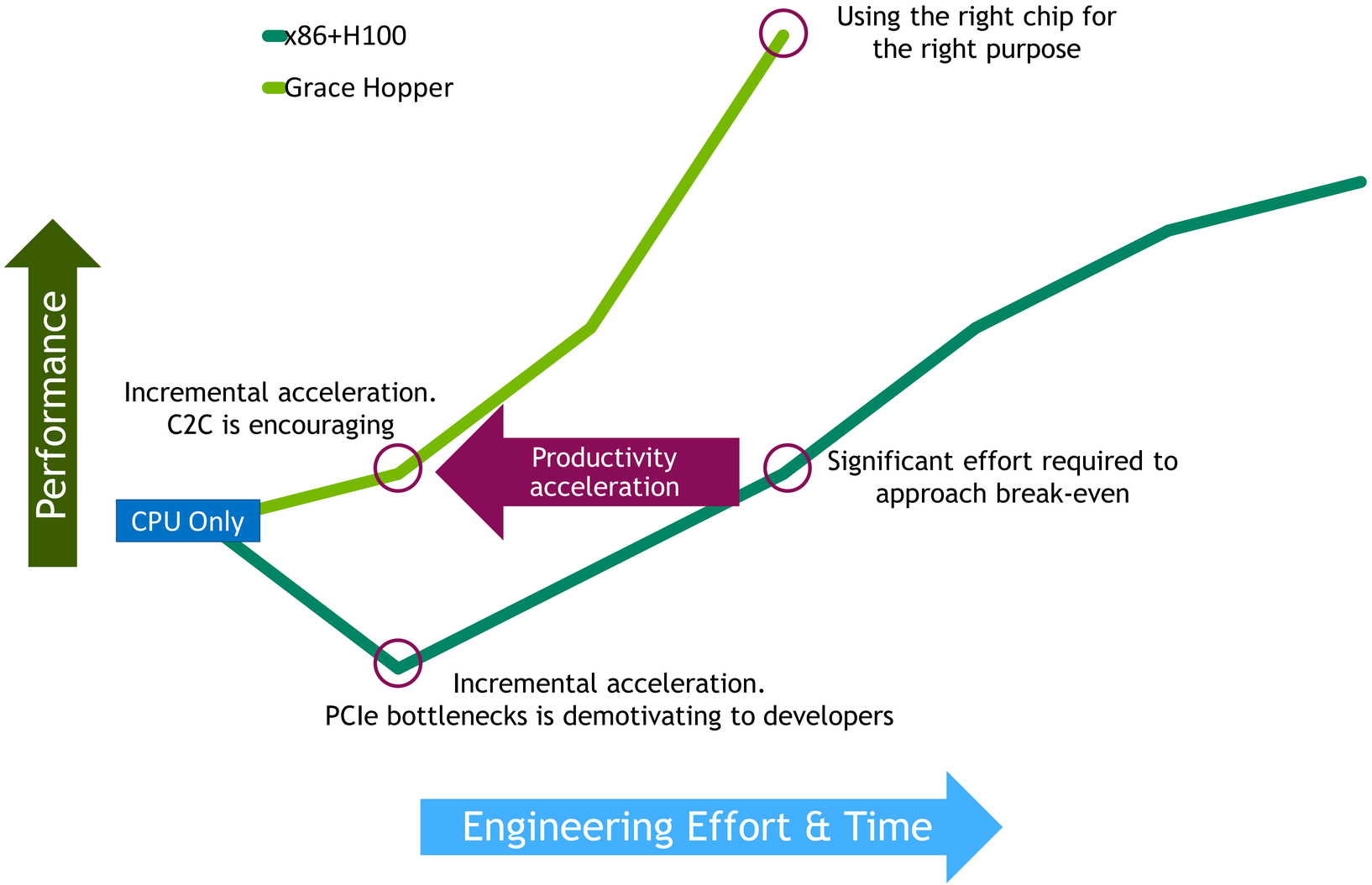
There can be ~~only~~ *more than* one.

Libraries	Standard Languages	Compiler Directives	CUDA Languages
• Accelerate common operations with little/no code changes. • Expert-tuned performance. • Forward support guarantees.	• Strong cross-platform support. • Single source code for multiple platforms. • Reduced learning curve.	• High cross-platform support. • Single source code for multiple platforms. • Reduced learning curve. • Additional programmer control.	• Exposes full GPU capabilities. • Trades portability for performance. • Distinct GPU/CPU code paths. • Full programmer control.
Programmer Productivity			Programmer Control

Approaches are interoperable.
Significantly increased productivity with coherent memory space.

Developer Velocity with Grace Hopper

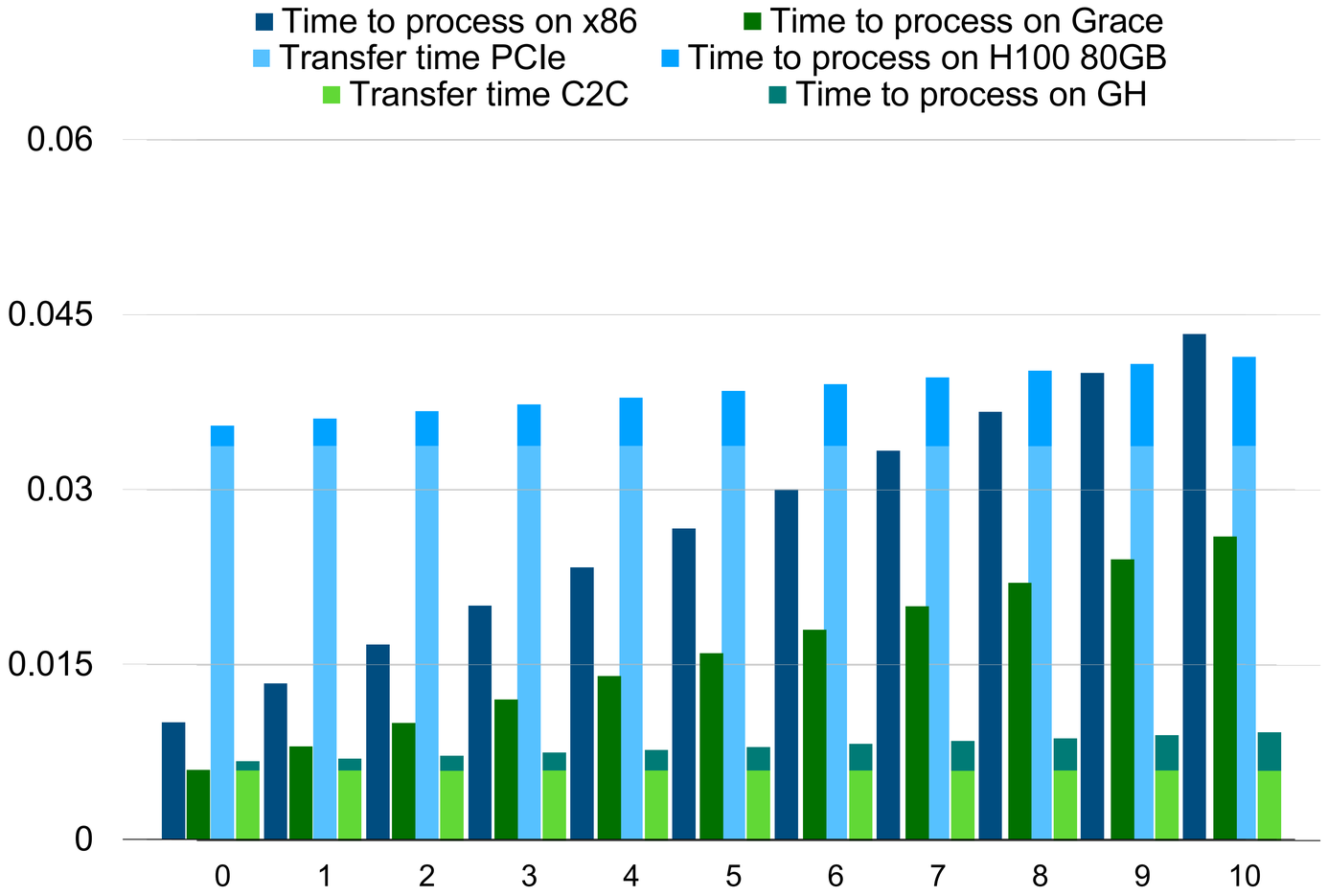
Accelerating the path to accelerated computing



Shifting the break-even point

Further lowering the barrier to GPU acceleration with C2C

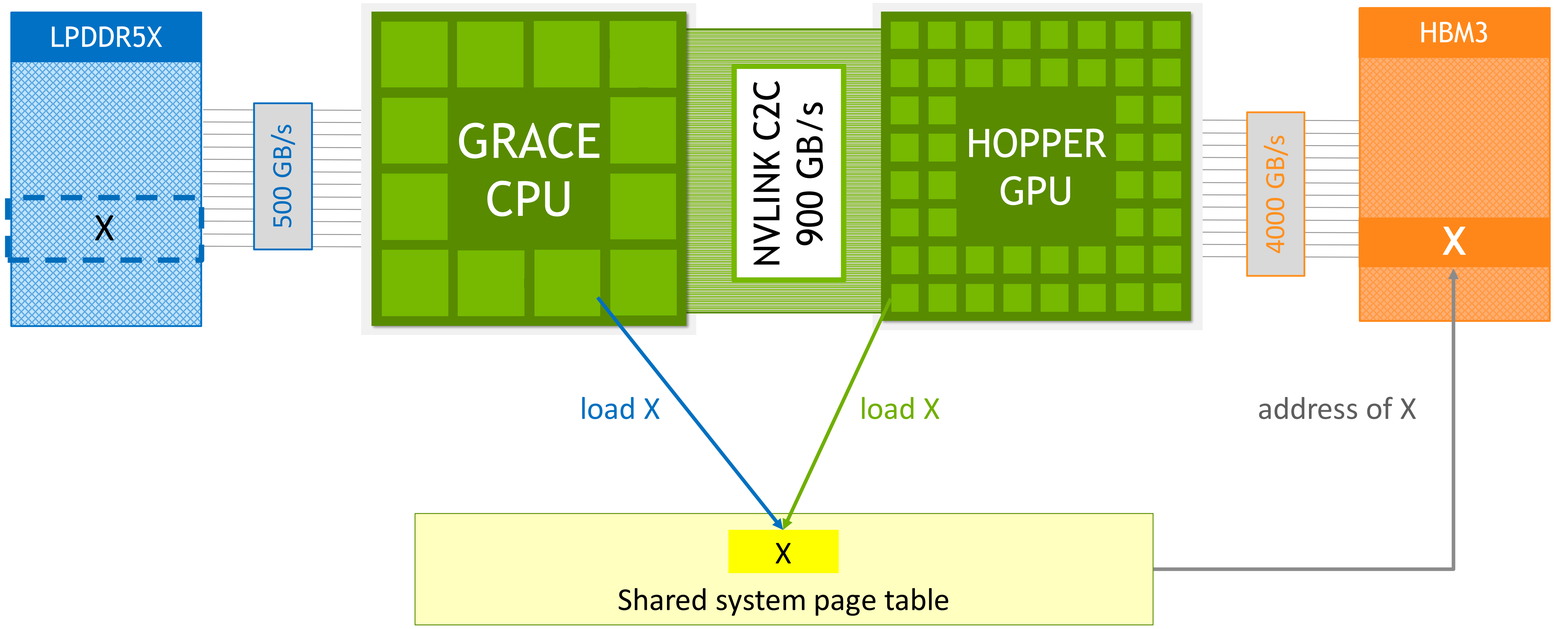
- Assume a memory-bandwidth bound workload
  - Idealized: time ~ data size / bandwidth
- Process 3+x GB of data on the CPU
- For GPU processing
  - Transfer 3 GB from / to GPU
  - Process 3+x GB of data on the GPU



High Bandwidth Memory Access & Automatic Data Migration

The system can automatically migrate
both managed and CPU-allocated memory
in order to optimize access speed

High Bandwidth Memory Access & Automatic Data Migration



ATS shared page table means that both CPU and GPU automatically access X in its new location after migration

NEMO Ocean Model

A partially accelerated case utilizing unified memory on Grace-Hopper

The "**N**ucleus for **E**uropean **M**odelling of the **O**cean" (**NEMO**) is a state-of-the-art modelling framework, used for research activities and forecasting services in ocean and climate sciences.

- **Setup (NEMO v4.2.0)**
  - **GYRE_PISCES** benchmark
    - Scaling factor for grid resolution: **nn_GYRE = 25**
      - ~ORCA ½ grid
      - ~80 GB RAM, fits on single GPU
  - **MPI-only**, single core to every MPI process for CPU runs

- **Incremental porting** on Grace-Hopper **(480GB)** using unified memory and access-counter based migrations
  - Memory management left to runtime – **system-allocated memory with automatic migrations**
    - compile with `-gpu=unified,nomanaged`
  - Simply offloading loops to GPU using **OpenACC**, in 3 steps:
    - **Horizontal (lateral) diffusion,**
    - **Advection,**
    - **Vertical diffusion and time-filtering,**

    for both “active” (**TRA**) and “passive” (**TRC**) tracer transport

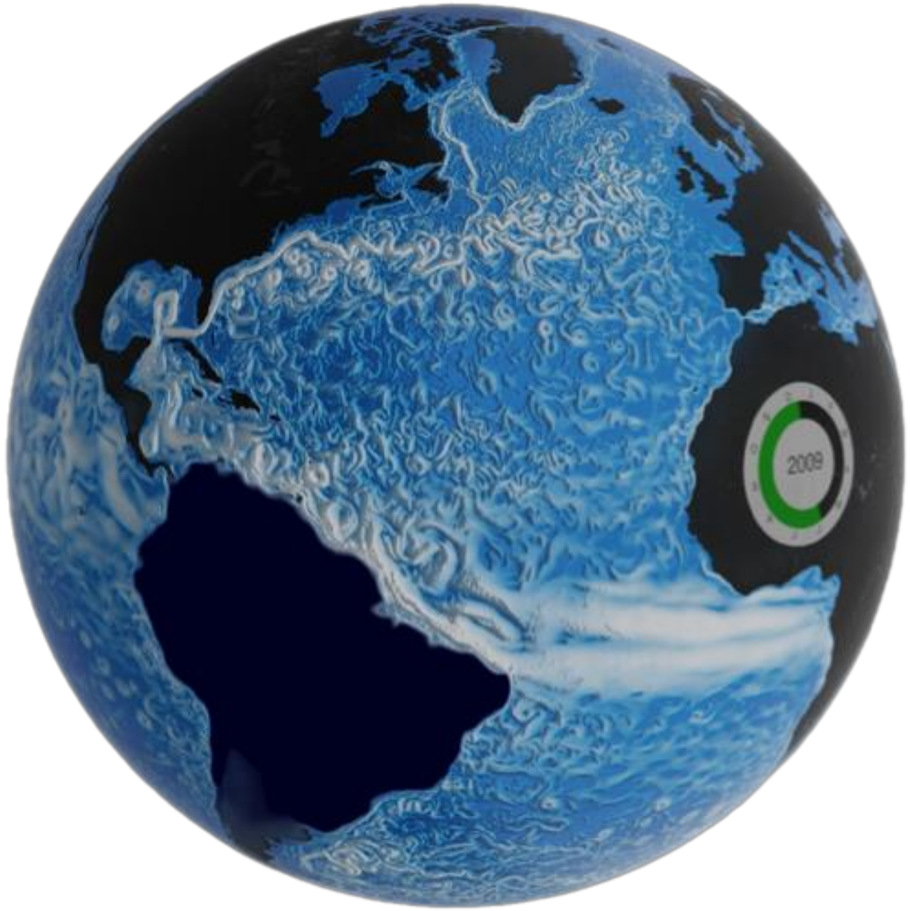
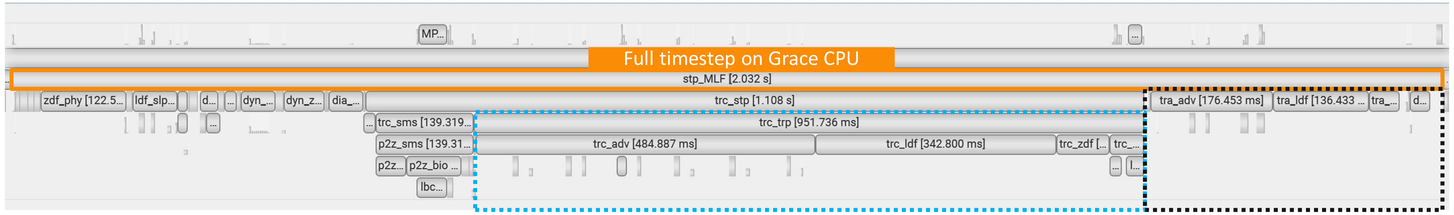


Image source:
NEMO User Guide — NEMO release-4.2.2 documentation (nemo-ocean.io)

Porting NEMO to Grace-Hopper using Unified Memory

Incremental porting, zooming in to a single timestep …



“Passive” tracer transport (TRC)

“Active” tracer transport (TRA)

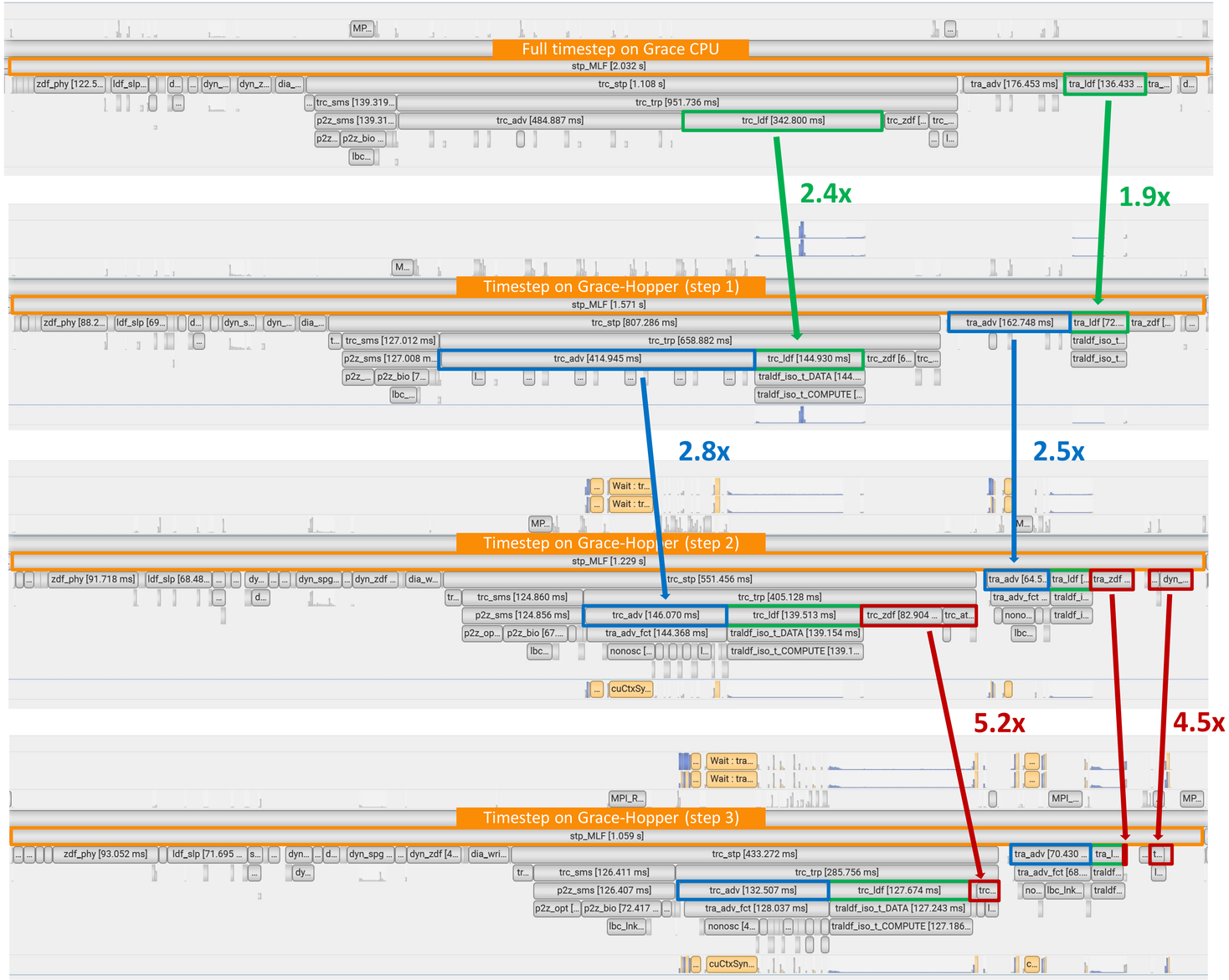
Porting NEMO to Grace-Hopper using Unified Memory

Incremental porting, zooming in to a single timestep …

Ported to GPU:

- **Horizontal diffusion**
- **Advection**
- **Vertical diffusion and time-filtering**

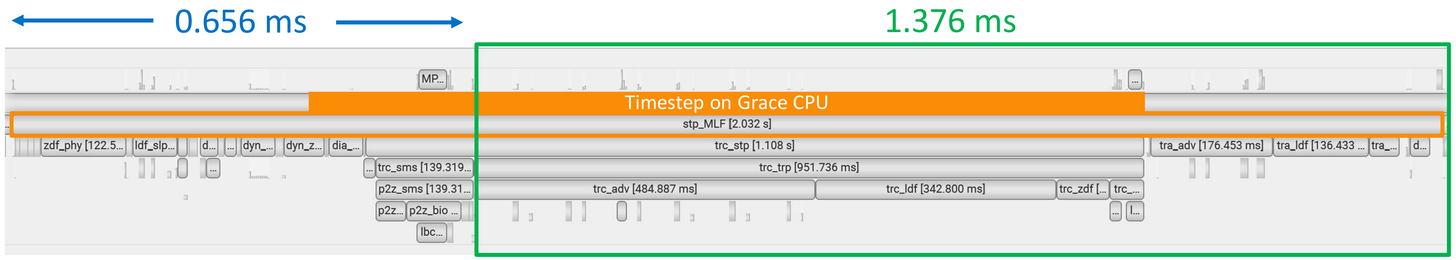
We run multiple (i.e. 40) MPI processes on **CPU and GPU** using **MPS**, and use **“migratable”** system allocated memory



Porting NEMO to Grace-Hopper using Unified Memory

A deeper look into the effect of access-counter based migrations on the partially accelerated port

CPU only run

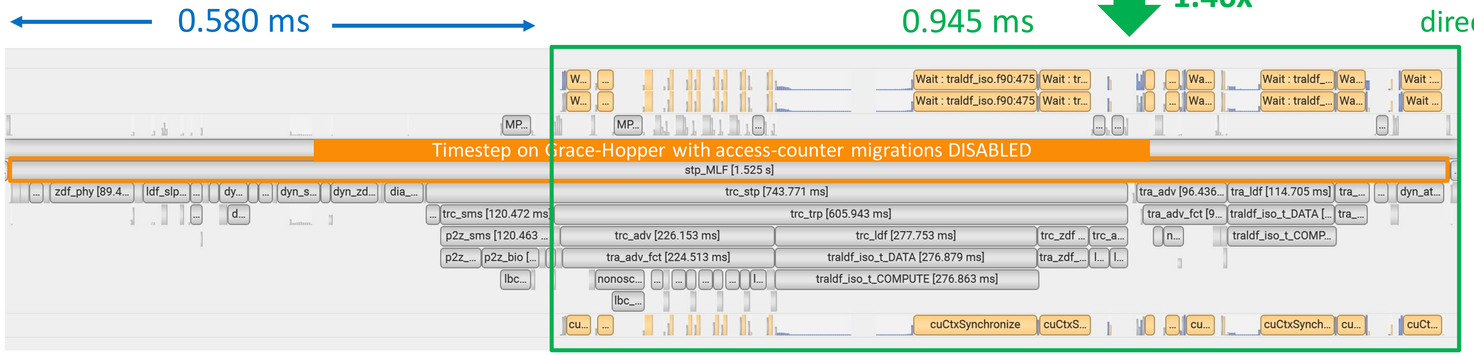


1.46x

1.33x

GPU kernels pull data first-touched by CPU directly from **CPU memory**

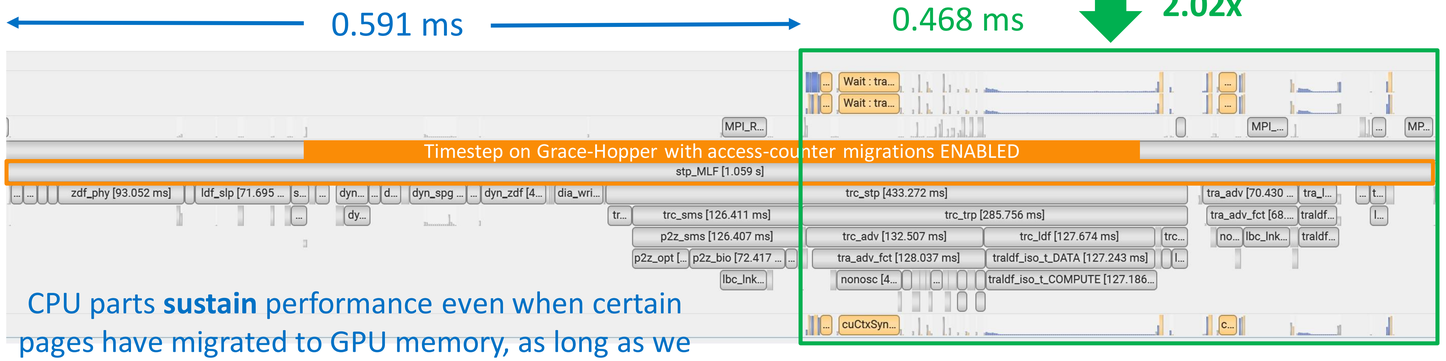
Tracer transport on GPU with migrations disabled
(buffers with first-touch on CPU will never migrate to GPU memory)



2.02x

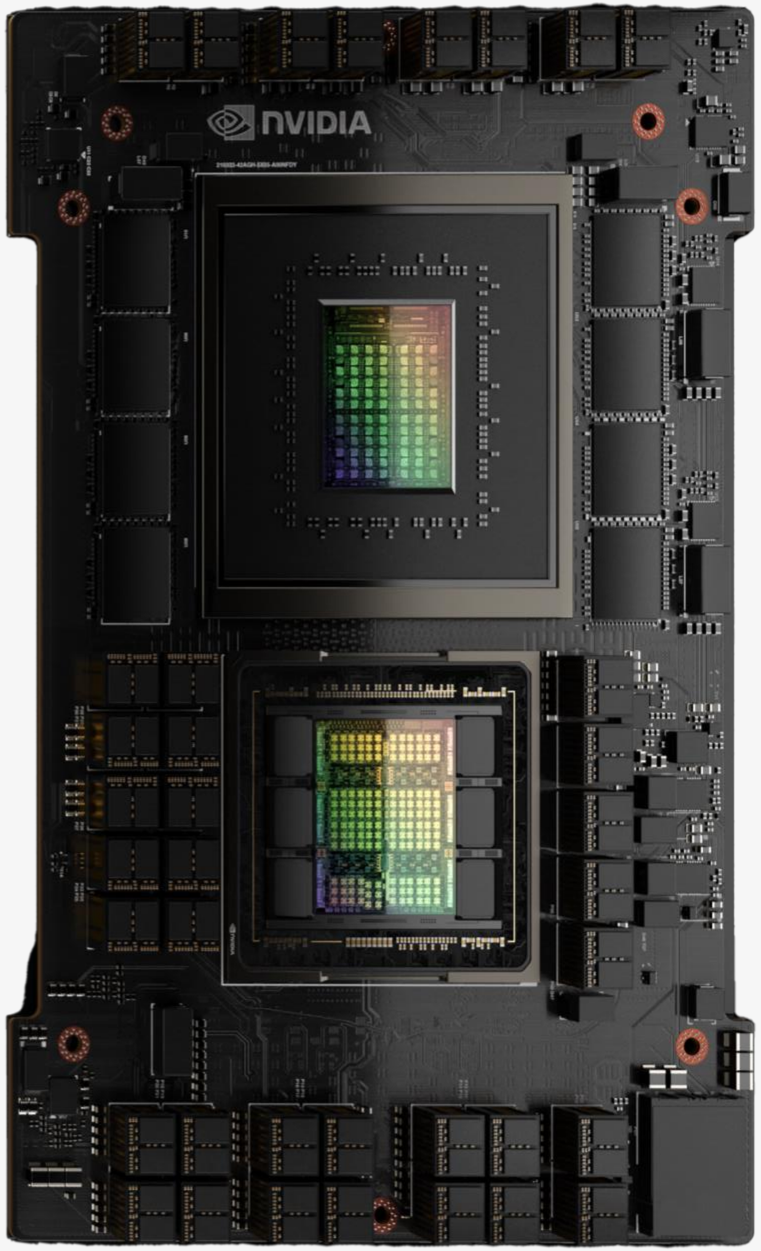
1.44x

Enabling automatic page migrations from CPU to GPU
(“hot” pages migrate to GPU)



CPU parts **sustain** performance even when certain pages have migrated to GPU memory, as long as we use enough processes to saturate C2C BW

GPU kernels become faster as more and more pages migrate to GPU



Science with Grace Hopper

Better than ever

- Balanced platform with
  - Hopper GPU
  - Grace CPU
  - C2C Interconnect (High Bandwidth)
  - Unified Memory Space
- Easier to accelerate
  - Widening bottlenecks unleashes performance
  - Easier to program
- Demonstrated speedups for multiple existing scientific workflows
- More codes being ported
- Multiple Installations coming online