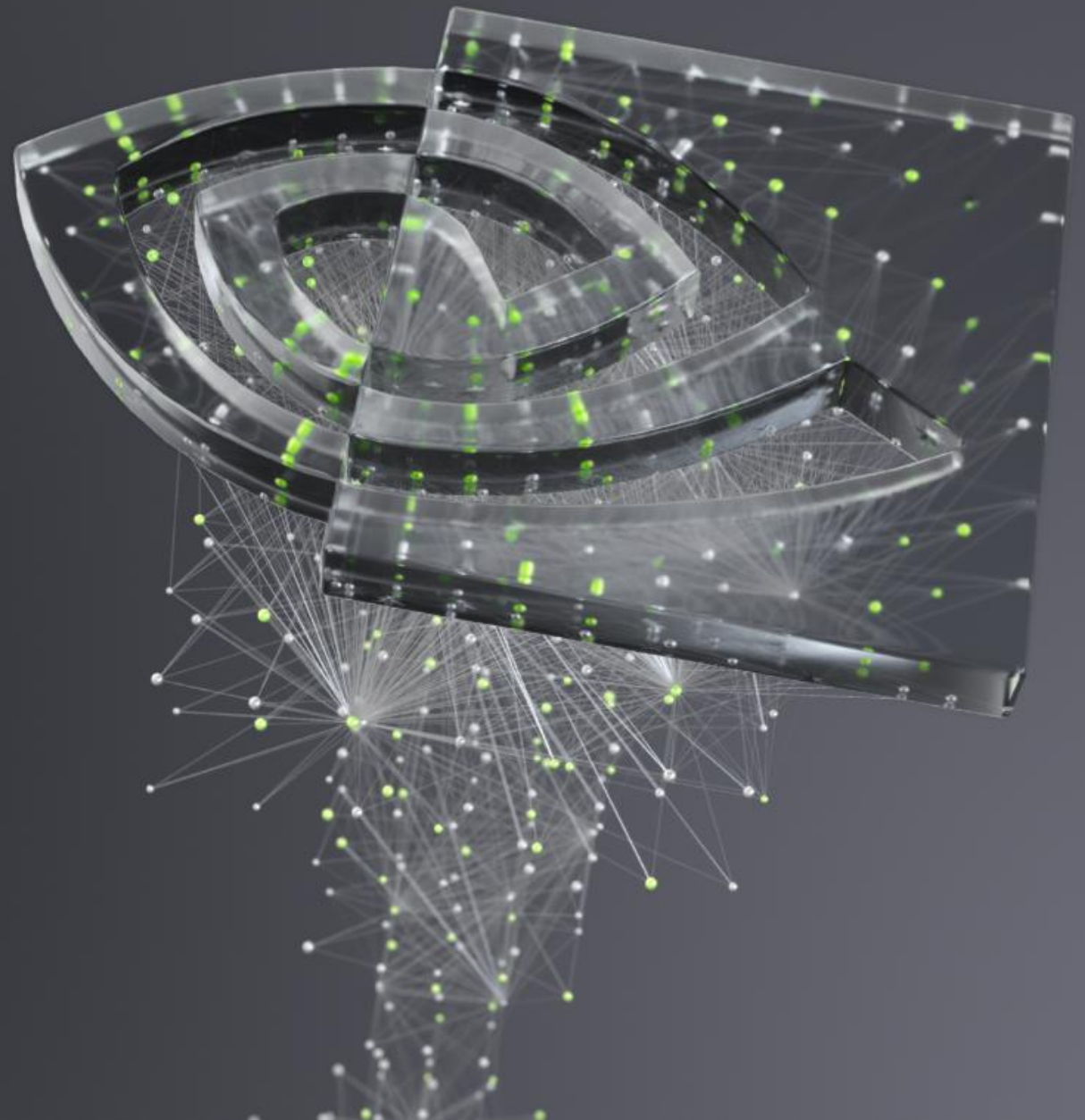


GRACE AND ANEXT UPDATE

NCHC, November 2021

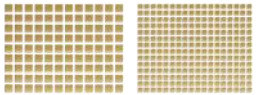
NVIDIA CONFIDENTIAL Presentation

Customer guidelines

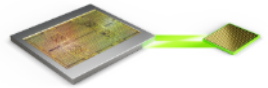
- All information in the following presentation is NVIDIA confidential, including codenames, future products, and performance projections
- No information in this presentation is allowed to be revealed or published without NVIDIA consent
- Sharing or distributing copies of this presentation to anyone is strictly prohibited
- Use of cameras & screen capture is strictly prohibited

CURRENTLY SHIPPING NVIDIA A100 40/80GB

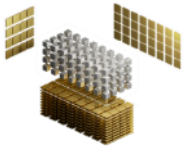
Supercharging The World's Highest Performing AI Supercomputing GPU



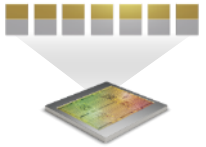
80GB HBM2e
For largest datasets and models



2TB/s +
World's highest memory bandwidth to feed the world's fastest GPU

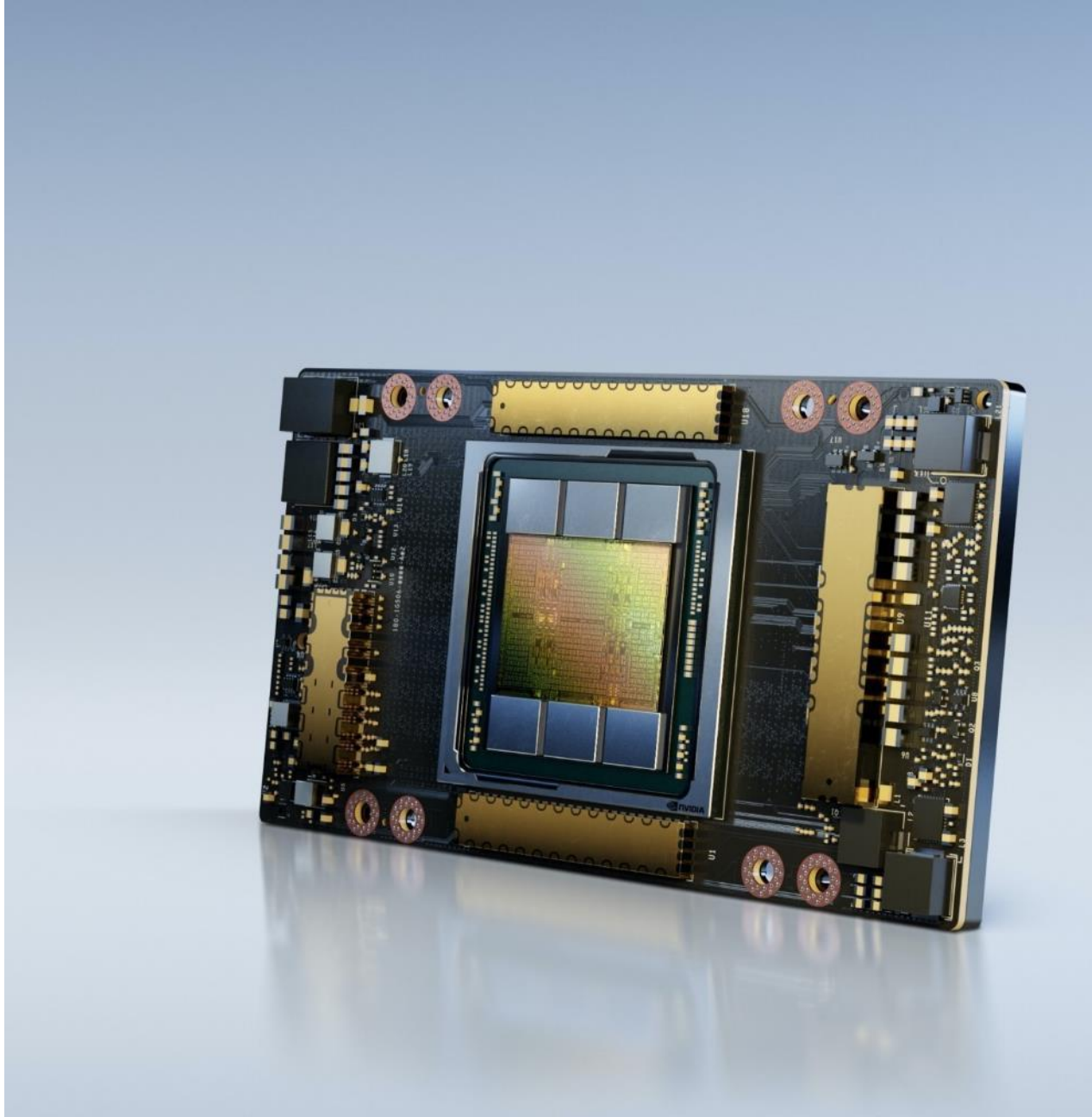


3rd Gen Tensor Core



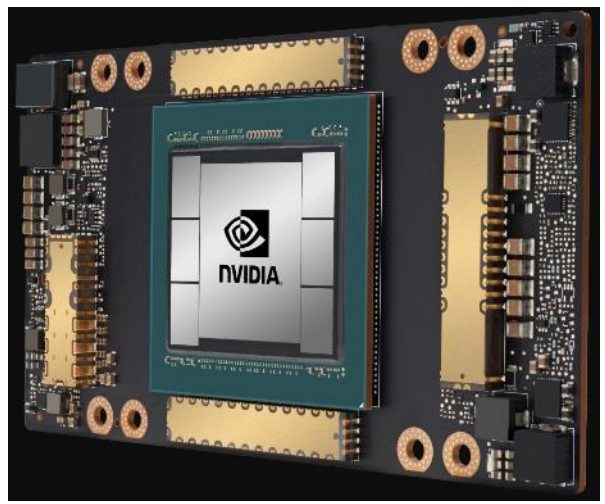
Multi-Instance GPU

3rd Gen NVLink



A100 AND V100 SPECS

	V100	A100
SMs	80	108
Tensor Core Precision	FP16	FP64, TF32, BF16, FP16, I8, I4, B1
Shared Memory per Block	96 kB	160 kB
L2 Cache Size	6144 kB	40960 kB
Memory Bandwidth	900 GB/sec	1555 GB/sec
NVLink Interconnect	300 GB/sec	600 GB/sec
FP64 Throughput	7.9 TFLOPS	9.7 | 19.5 TFLOPS
TF32 Tensor Core	N/A	156 | 312 TFLOPS



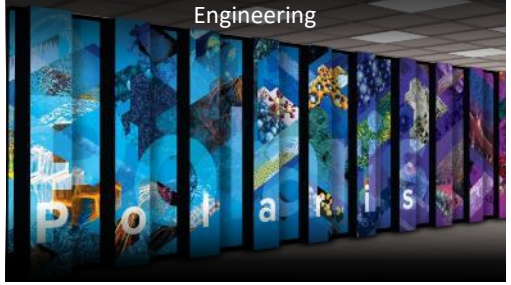
ACCELERATED AI SUPERCOMPUTING MOMENTUM

70% of World's Fastest Systems
8 of Top 10 Supercomputers

TOP 500 The List.

NVIDIA

ARGONNE "POLARIS"
Leadership System for AI, Science, Engineering



NERSC "PERLMUTTER"
AI for Scientific Discovery

JŰLICH JUWELS
HPC Simulation & Research

DGX SUPERPODS
Industrial, Cloud-Native AI

MICROSOFT AZURE
On-Demand Supercomputing

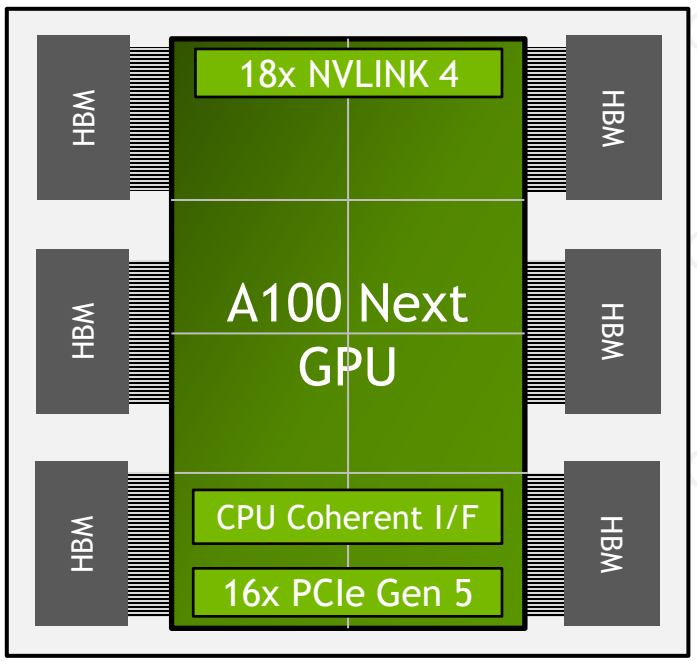
"ALPS" / "LEONARDO"
Exascale AI Supercomputers

TESLA AUTOMOTIVE
Industrial AI Supercomputer


ANEXT GPU

ANEXT GPU

Highest Performance, Efficiency & Utilization

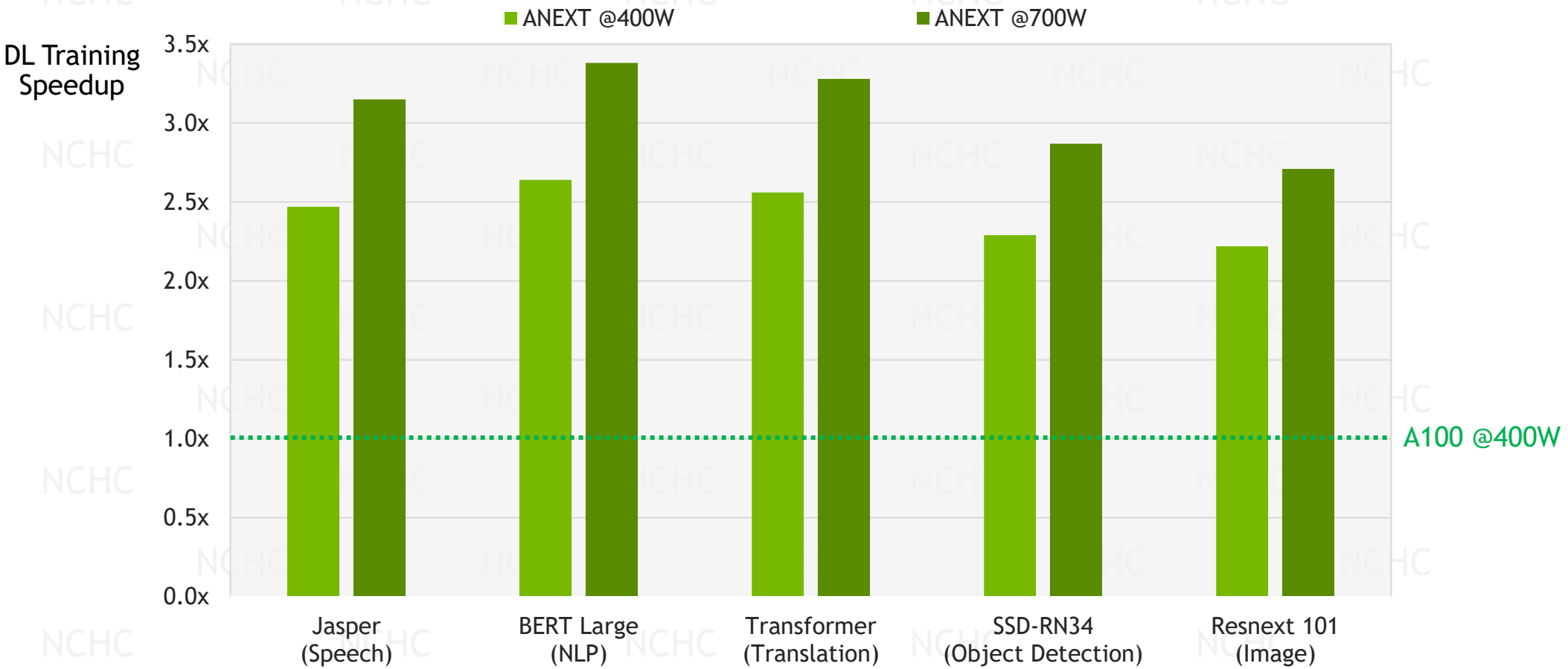


Technology	Capability
New Tensor Core Architecture	~2-3x FLOPS vs A100
Enhanced HBM3 Memory	~2-3x Bandwidth vs A100 Bandwidth boost from data compression
Multi-Instance GPU (MIG)	Up to 8 Instances
NVLINK 4	1.5x A100; 450GB/s/direction
PCIE	PCIE Gen5 x16
Chip to Chip Interface	450GB/s/direction CPU-GPU Coherent with NV ARM
GPU TDP	~700W

Preliminary disclosure, subject to change
Max chip capability shown, product SKUs specs can be different

>3X FASTER WITH FP32 TRAINING

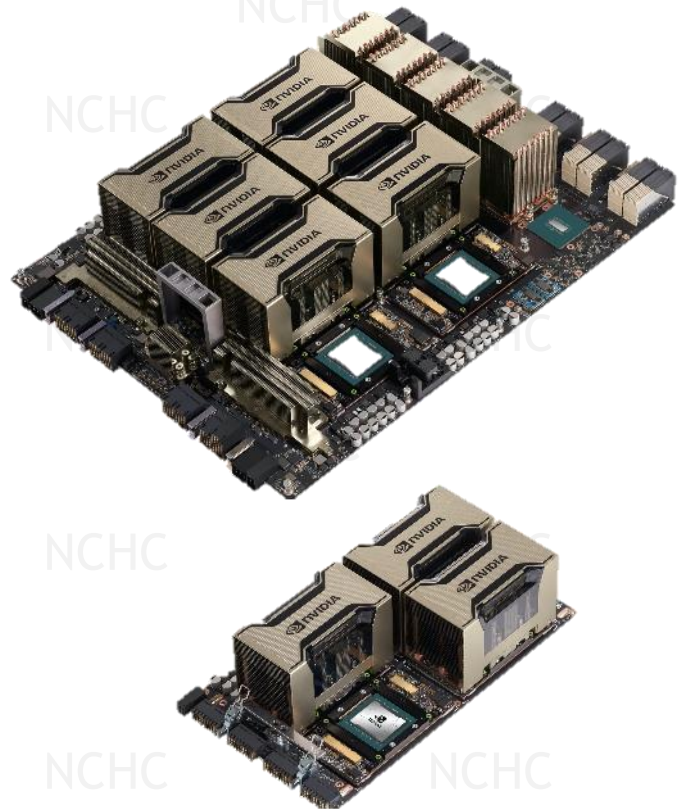
700W TDP GIVES CLOSE TO 1.3X BOOST OVER 400W



Note: Projection is preliminary and subject to change

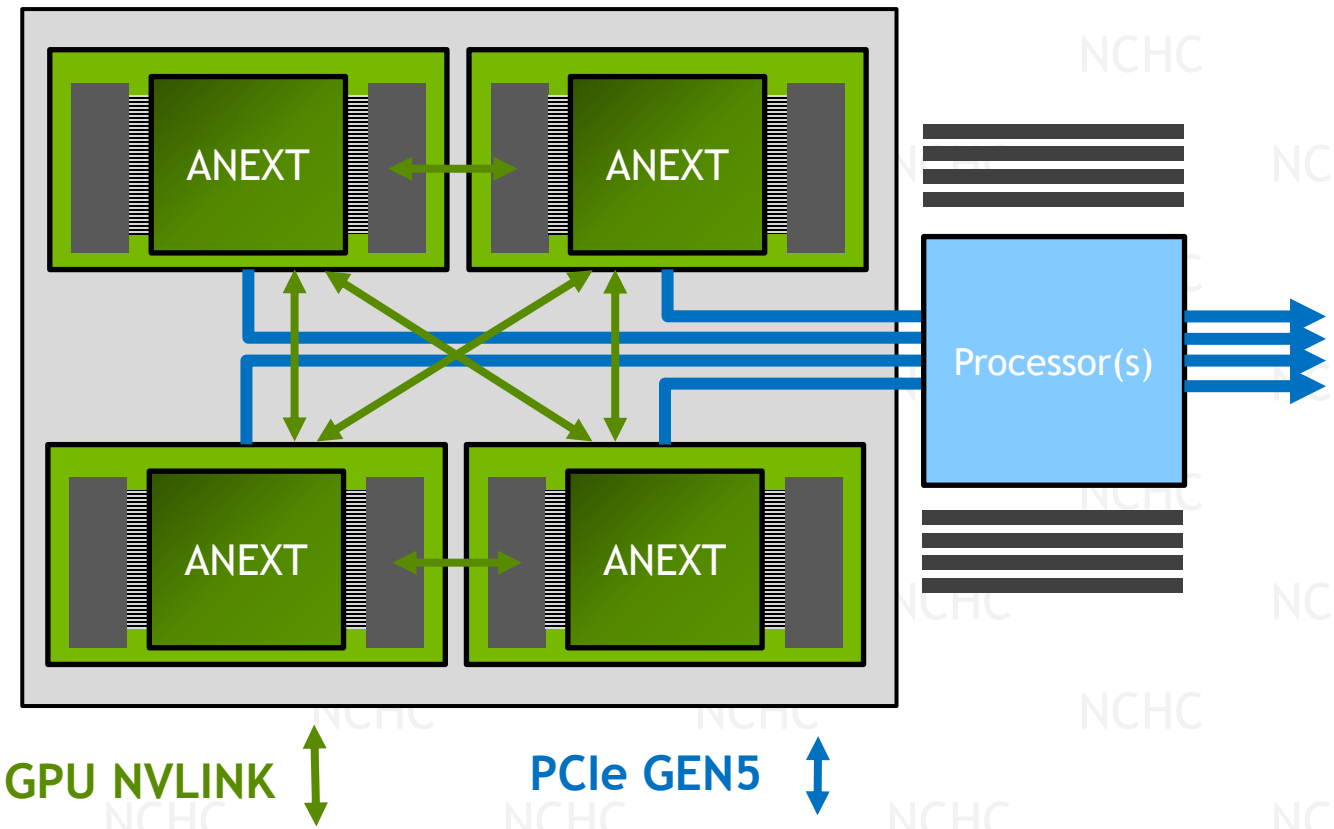
ANEXT HGX PLATFORM

ANEXT GPU	Attributes
SOUL	Highest AI & HPC Performance Multi-Instance GPU to maximize utilization
FORM FACTOR	Delta-Next 8-ANEXT with NVSWITCH Redstone-Next 4-ANEXT with direct connect
ARCHITECTURE	New Tensor Core design HBM3 Enhanced Multi-Instance GPU architecture
I/O	PCIe Gen5 New NVSWITCH Higher speed NVLINK
GPU TDP	HGX: 600W+ PCIe Card: 300W+



EXAMPLE HGX ARCHITECTURE FOR HPC

What Runs Best Here?

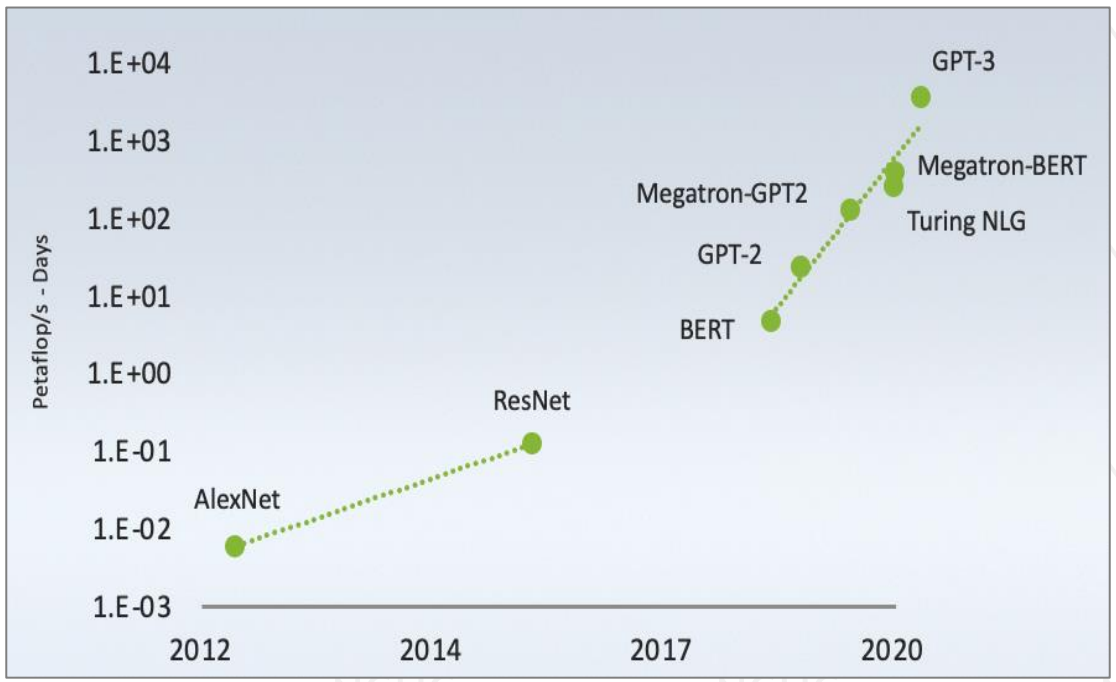


- HPC applications with 90% or greater “accelerator friendly” content
- AI Training - Can handle up to 320GB models in fast memory (640GB for the 8-way)

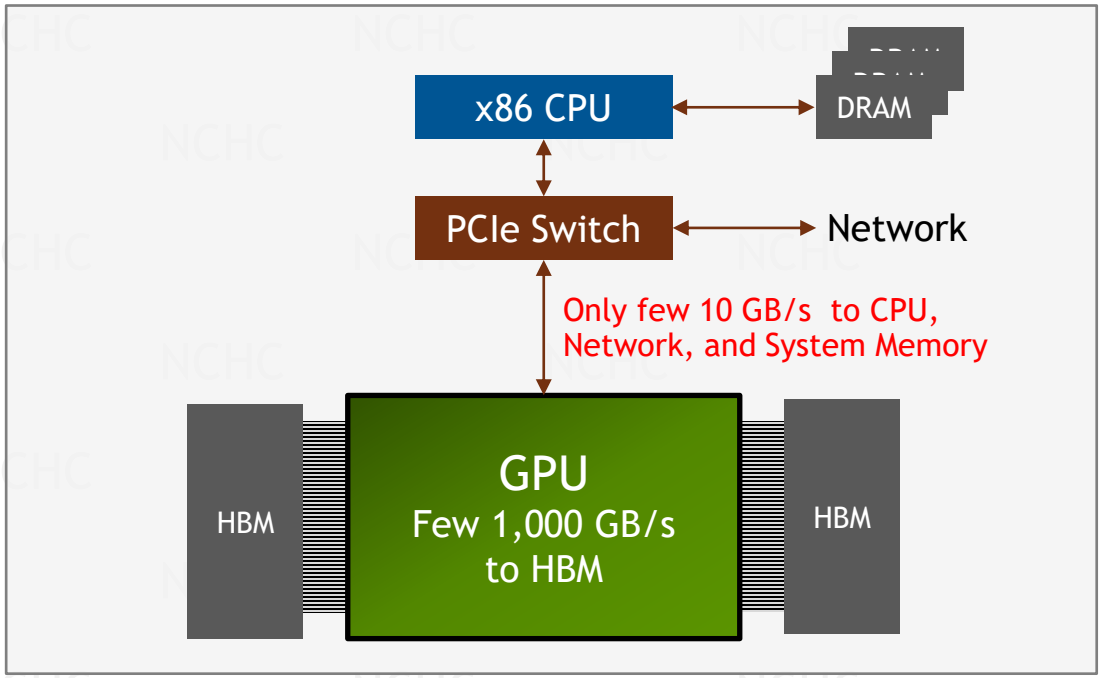
WHY ARM?

GIANT MODELS PUSH THE LIMITS OF EXISTING ARCHITECTURES

Exponential growth
30,000X in 5 Years | Now 2x Every 2.5 Months



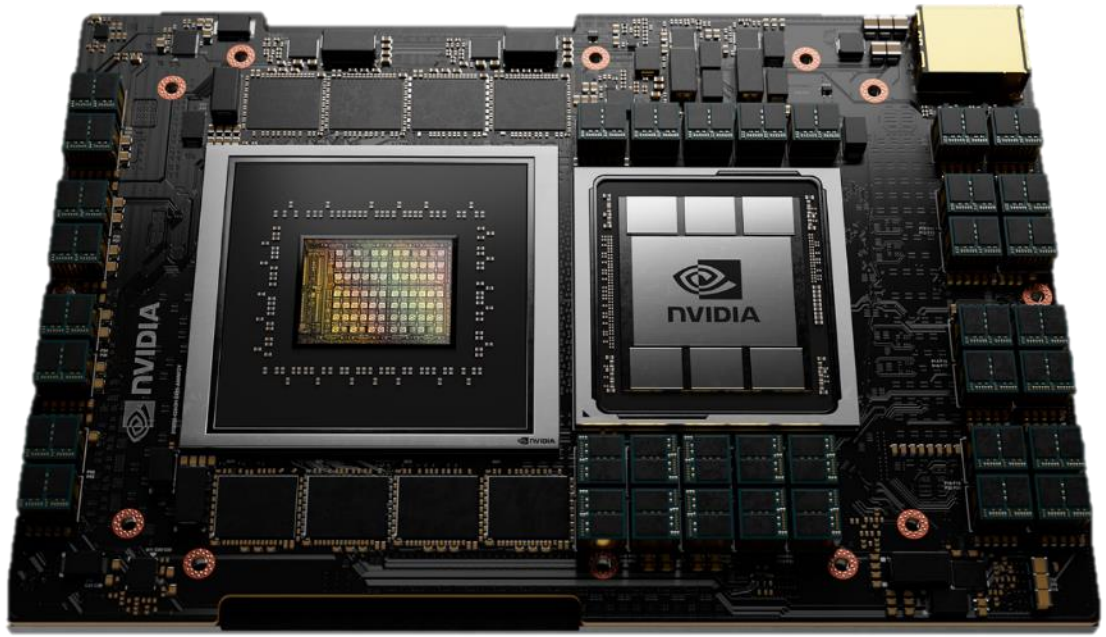
B/W Bottlenecks
Insufficient single threaded performance



Preliminary disclosure, subject to change. Max chip capability shown, product SKUs specs can be different

ANNOUNCING NVIDIA GRACE

Breakthrough CPU Designed for Giant-Scale AI and HPC Applications



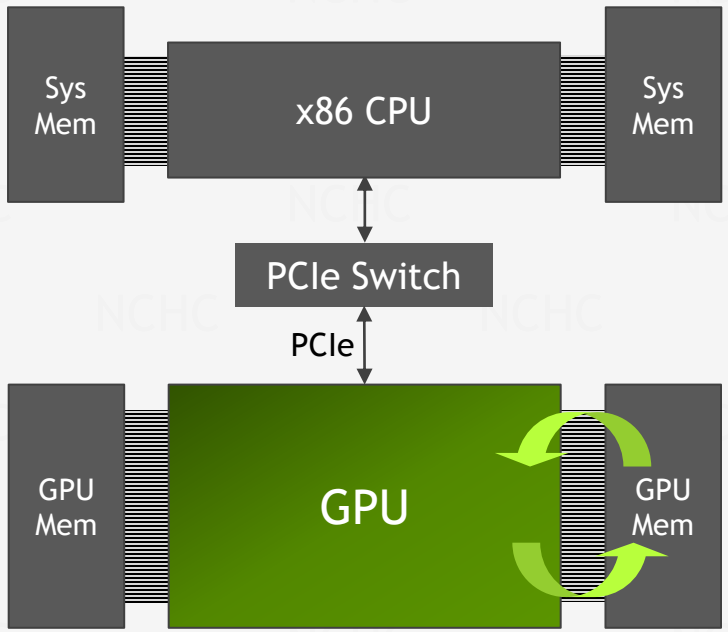
FASTEST INTERCONNECTS
>900 GB/s Cache Coherent NVLink CPU To GPU (14x)

HIGHEST MEMORY BANDWIDTH
>500GB/s LPDDR5x w/ ECC
>2x Higher B/W
10x Higher Energy Efficiency

NEXT GENERATION ARM NEOVERSE CORES
>300 SPECrate2017_int_base est.
Availability 2023

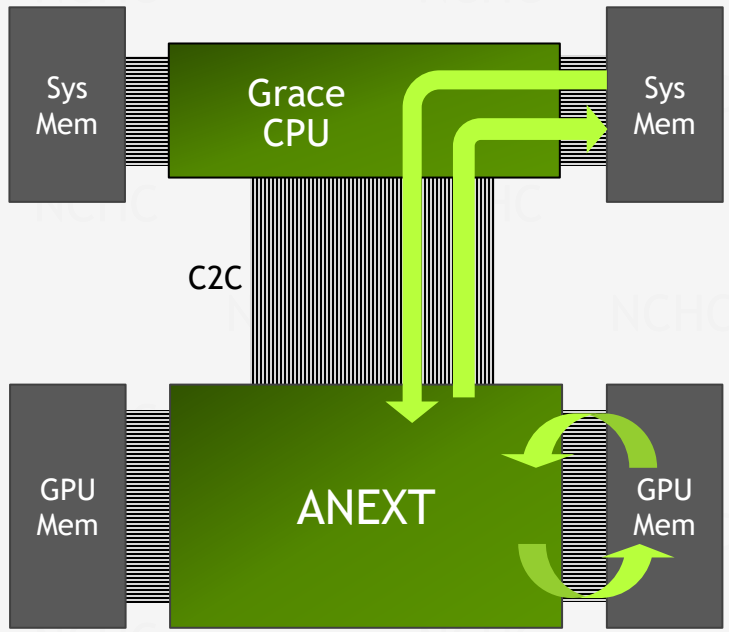
EXAMPLE USE CASE
GRACE ENABLE NEW “TENSOR OFFLOADING” TECHNIQUE

Traditional x86 + GPU Architecture



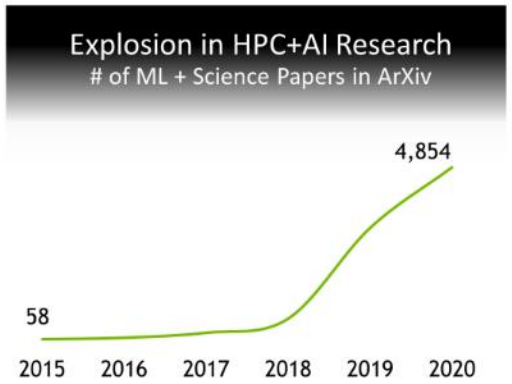
1) During training, GPU computes new tensor and store to GPU memory for fast read back
2) For large model, GPU memory size limits the maximum batch size that can be used

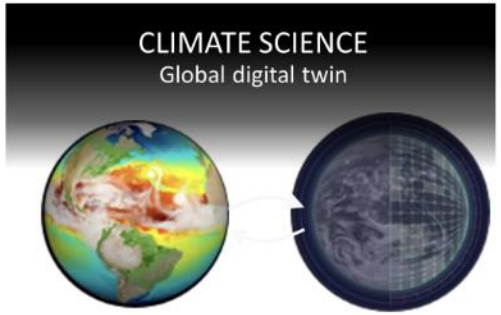
NV ARM + ANEXT Architecture

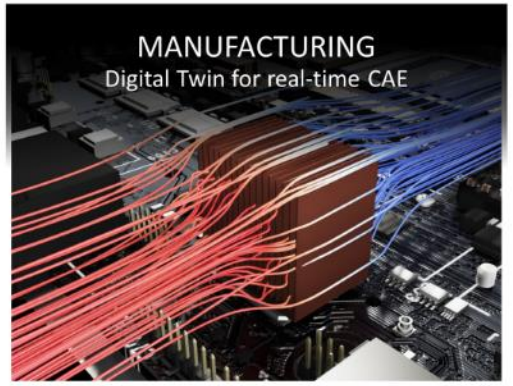


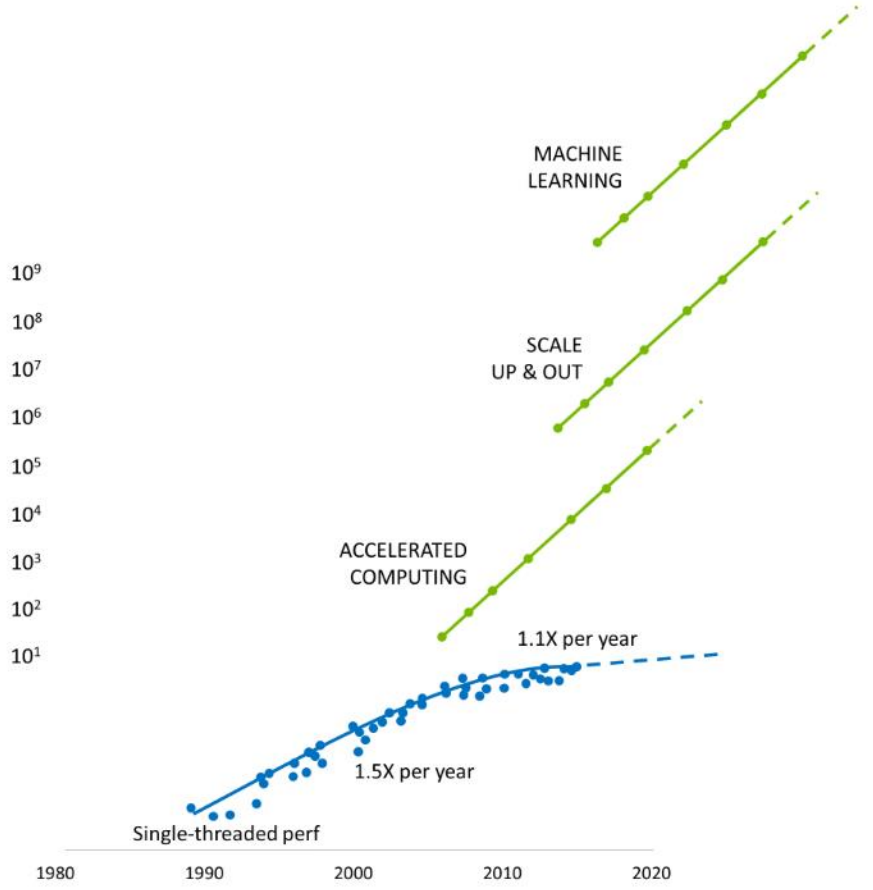
3) With new C2C bus, GPU can now offload tensor to system memory at high speed
4) Enabling larger batch size and hence more efficient use of the GPU

HPC AND AI ARE TRANSFORMING SIMULATION









Pushing the limit of molecular dynamics with *ab initio* accuracy to 100 million atoms with machine learning

Weile Jia*, Han Wang†, Mohan Chen‡, Denghui Lu‡, Lin Lin*¶, Roberto Car‖, Weinan E‖, Linfeng Zhang‖ ‡
*University of California, Berkeley, Berkeley, USA
Email: jiaweile@berkeley.edu, linlin@math.berkeley.edu
† Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, China
Email: wang_han@iapcm.ac.cn
‡CAPT, HEDPS,College of Engineering, Peking University, Beijing, China
Email: mohanchen@pku.edu.cn, denghuilu@pku.edu.cn
¶Lawrence Berkeley National Laboratory, Berkeley, USA
‖ Princeton University, Princeton, USA
Email: rcar@princeton.edu, weinan@math.princeton.edu,linfengz@princeton.edu

***Abstract*—For 35 years, *ab initio* molecular dynamics (AIMD) has been the method of choice for modeling complex atomistic phenomena from first principles. However, most AIMD applications are limited by computational cost to systems with thousands of atoms at most. We report that a machine learning-based simulation protocol (Deep Potential Molecular Dynamics), while retaining *ab initio* accuracy, can simulate more than 1 nanosecond-long trajectory of over 100 million atoms per day, using a highly optimized code (GPU DeePMD-kit) on the Summit supercomputer. Our code can efficiently scale up to the entire Summit supercomputer, attaining 91 PFLOPS in double precision (45.5% of the peak) and 162/275 PFLOPS in mixed-single/half precision. The great accomplishment of this work is that it opens the door to simulating unprecedented size and time scales with *ab initio* accuracy. It also poses new challenges to the next-generation supercomputer for a better integration of machine learning and physical modeling.**

***Index Terms*—Deep potential molecular dynamics, *ab initio* molecular dynamics, machine learning, GPU, heterogeneous architecture, Summit**

I. Justification for Prize

Record molecular dynamics simulation of >100 million atoms with *ab initio* accuracy. Double/mixed-single/mixed-half precision performance of 91/162/275 PFLOPS on 4,560 nodes of Summit (27,360 GPUs). For a 127-million-atom copper system, time-to-solution of $8.1/4.6/2.7\times10^{-10}$ s/step/atom, or equivalently 0.8/1.5/2.5 nanosecond/day, >1000× improvement w.r.t state-of-the-art.

II. Performance Attributes

Performance attribute	Our submission
Category of achievement	Time-to-solution, scalability
Type of method used	Deep potential molecular dynamics
Results reported on basis of	Whole application including I/O
Precision reported	Double precision, mixed precision
System scale	Measured on full system
Measurements	Timers, FLOP count

§Corresponding author

III. Overview of the Problem

A. *ab initio* molecular dynamics

Molecular dynamics (MD) [1], [2] is an *in silico* simulation tool for describing atomic processes that occur in materials and molecules. The accuracy of MD lies in the description of atomic interactions, for which the *ab initio* molecular dynamics (AIMD) scheme [3], [4] stands out by evolving atomic systems with the interatomic forces generated on-the-fly using first-principles electronic structure methods such as the density functional theory (DFT) [5], [6]. AIMD permits chemical bond cleavage and formation events to occur and accounts for electronic polarization effects. Due to the faithful description of atomic interactions by DFT, AIMD has been the major avenue for the microscopic understanding of a broad spectrum of issues, such as drug discovery [7], [8], complex chemical processes [9], [10], nanotechnology [11], etc.

The computational cost of AIMD generally scales cubically with respect to the number of electronic degrees of freedom. On a desktop workstation, the typical spatial and temporal scales achievable by AIMD are ~100 atoms and ~10 picoseconds. From 2006 to 2019, the peak performance of the world's fastest supercomputer has increased about 550-folds, (from 360 TFLOPS of BlueGene/L to 200 PFLOPS of Summit), but the accessible system size has only increased 8 times (from 1K Molybdenum atoms with 12K valence electrons [12] to 11K Magnesium atoms with 105K valence electrons [13]), which obeys almost perfectly the cubic-scaling law. Linear-scaling DFT methods [14]–[17] have been under active developments, yet the pre-factor in the complexity is still large, and the time scales attainable in MD simulations remain rather short.

For problems in complex chemical reactions [18], [19], electrochemical cells [20], nanocrystalline materials [21], [22], radiation damage [23], dynamic fracture, and crack propagation [24], [25], etc., the required system size typically ranges

SC20, November 9-19, 2020, Is Everywhere We Are
978-1-7281-9998-6/20/$31.00 ©2020 IEEE

GORDON BELL 2020 AWARD

Challenge

Demonstrate Ab Initio Accuracy at scales that can only be modeled with Molecular Dynamics Accuracy using conventional Molecular Dynamics methods

Solution

DeePKit MD refactored the DP DNN that previously was only running on one GPU and demonstrated at scale on SUMMIT with TensorFlow and LAMMPS

127 Mn Atom Copper Model achieved 2.5 ns/day

679 Mn Atom Water Model achieved 42 ns/day

Impact

DFT accuracy that is 3 Orders of Magnitude faster that previous best in class demonstration

PRINCETON UNIVERSITY

ALPHAFOLD-2

nature

NEWS | 30 November 2020

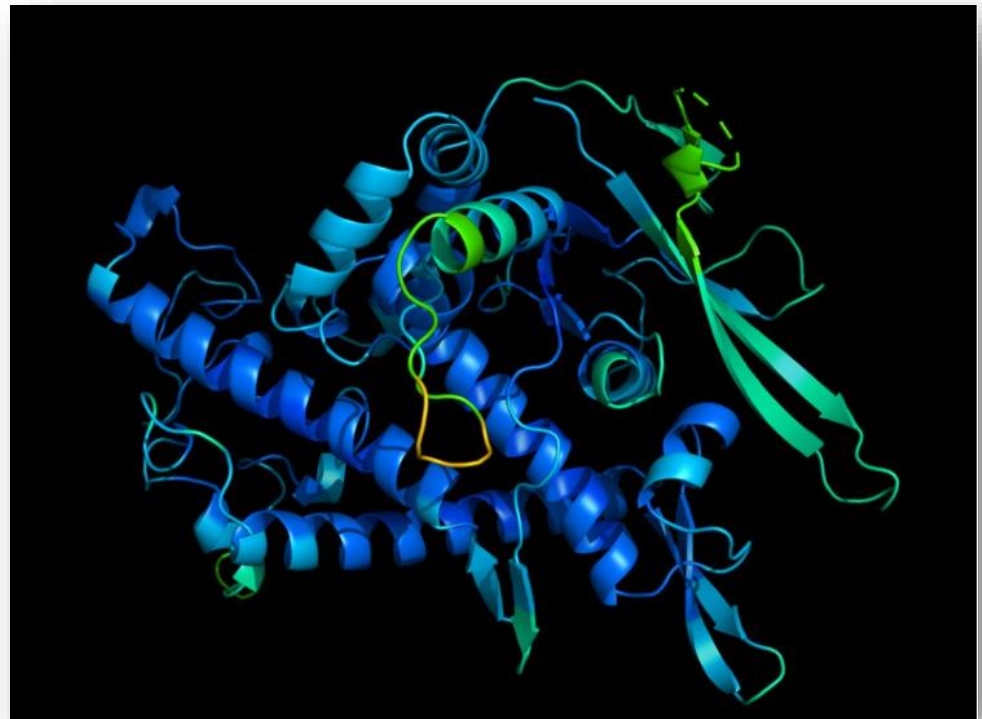
‘It will change everything’: DeepMind’s AI makes gigantic leap in solving protein structures

Google’s deep-learning program for determining the 3D shapes of proteins stands to transform biology, say scientists.

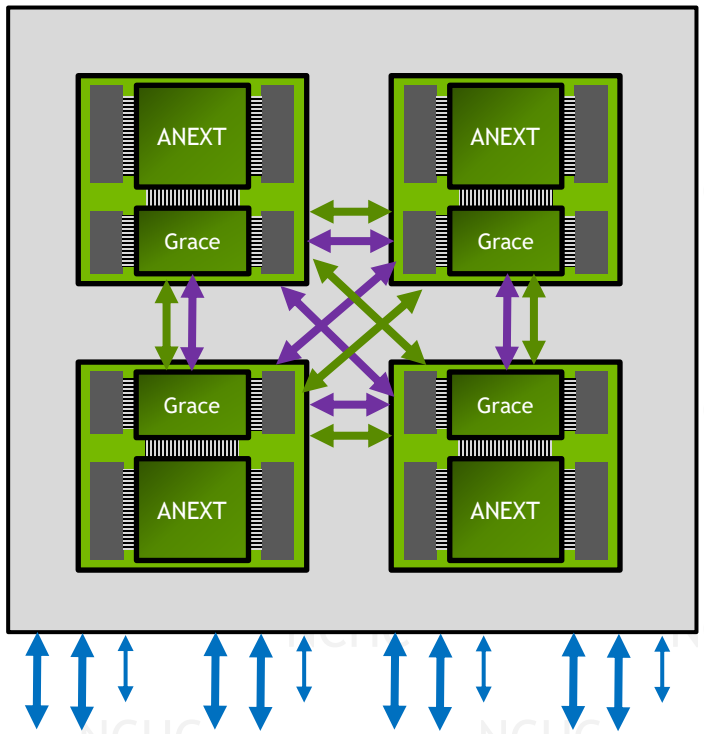
NEWS | 15 July 2021

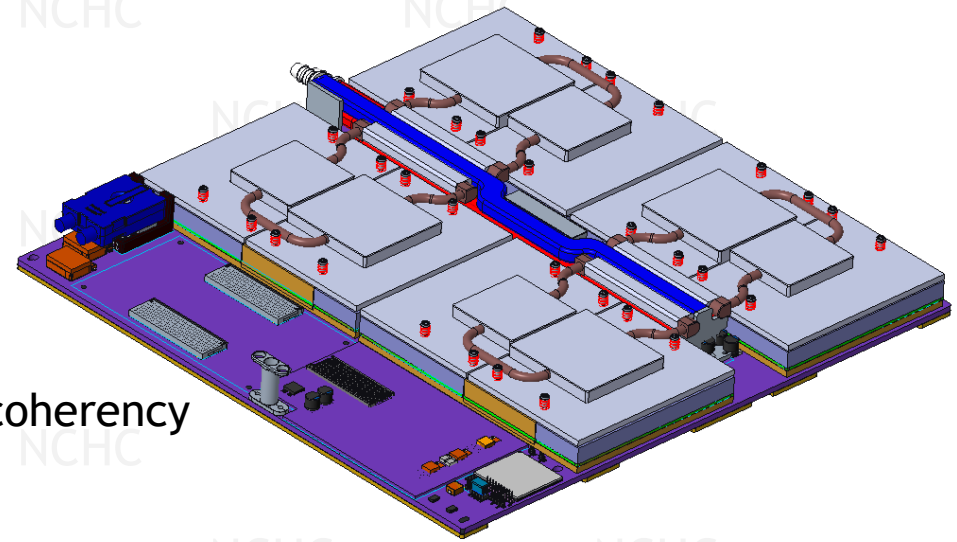
DeepMind’s AI for protein structure is coming to the masses

Machine-learning systems from the company and from a rival academic group are now open source and freely accessible.



4-WAY GRACE-ANEXT NODE





- Hardware managed coherency between Quad CPU
- Up to ~2TB of CPU memory
- Up to 2 PCIe Gen5 NICs per ANEXT with full GPU RDMA
  - Direct PCIe to GPU Mem, no staging at CPU Mem

Preliminary disclosure, subject to change

THE WORLD'S FASTEST SUPERCOMPUTER FOR AI

20 Exaflops of AI

Accelerated w/ NVIDIA Grace CPU and NVIDIA GPU

HPC and AI For Scientific and Commercial Apps

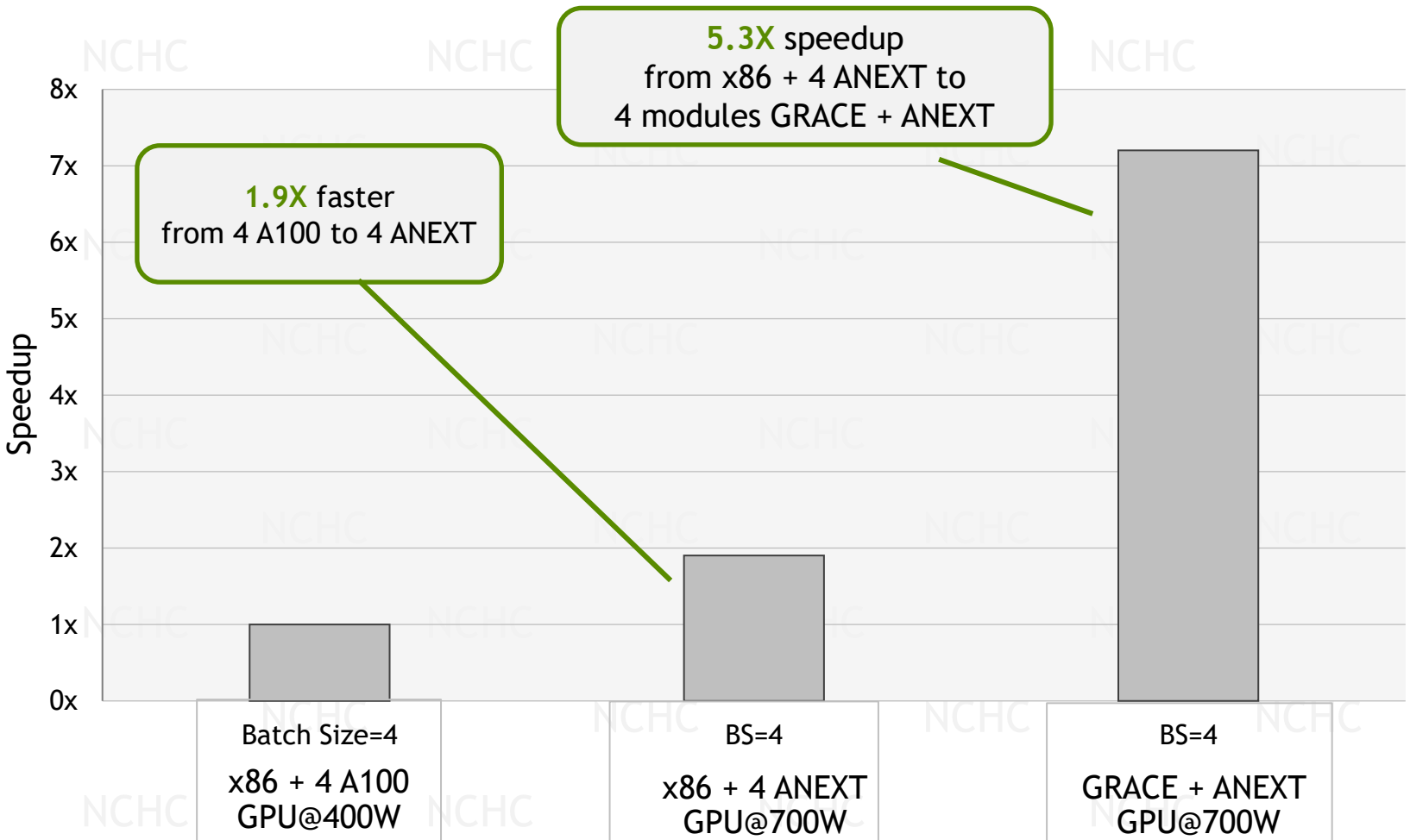
Advance Weather, Climate, and Material Science

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Hewlett Packard Enterprise

NVIDIA

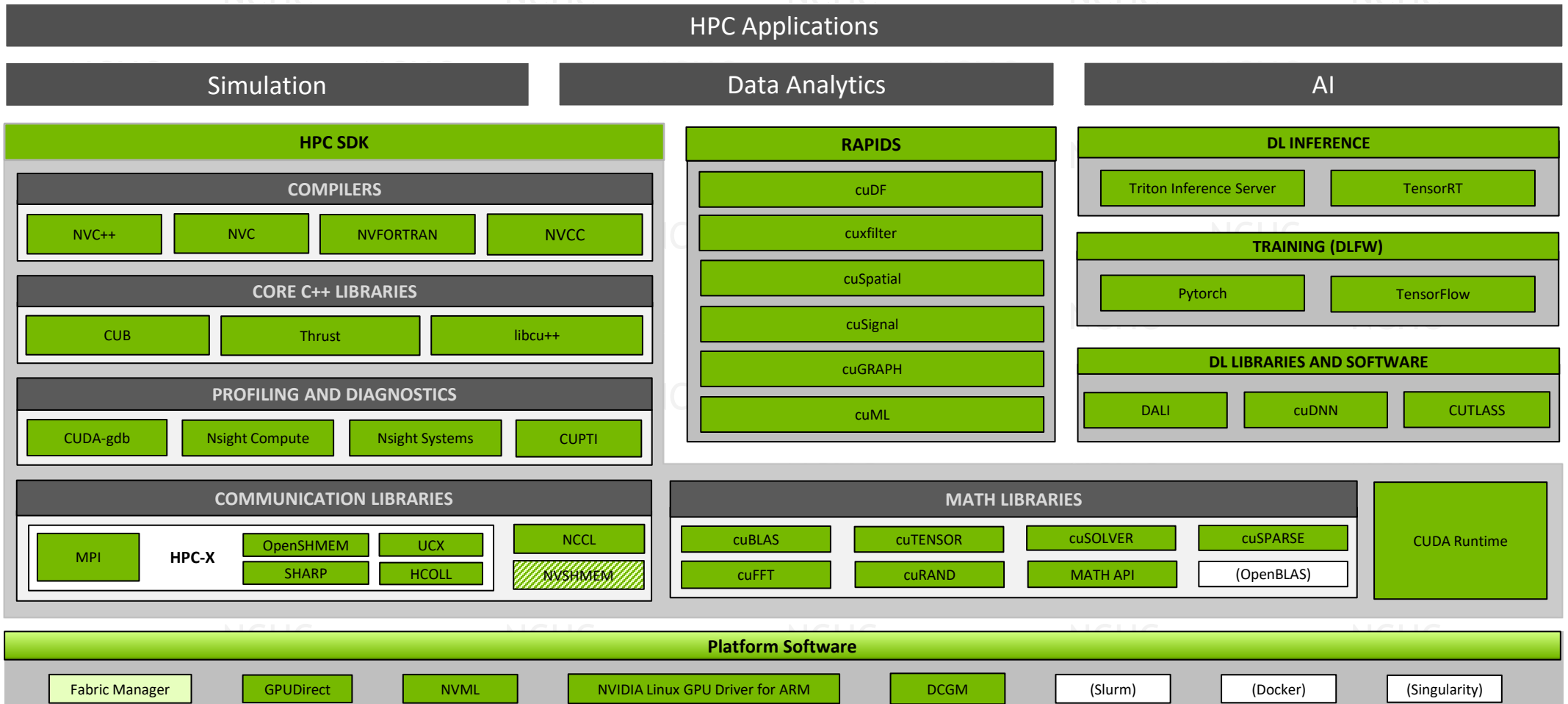
145B NLP MODEL FINETUNING



- ANEXT offloads tensor to GRACE's memory to increase batch size, enabling large perf gain
- ANEXT can also offload tensor to x86's memory, but the perf becomes slower due to PCIe bandwidth bottleneck

Projection based on architecture simulation, subject to change

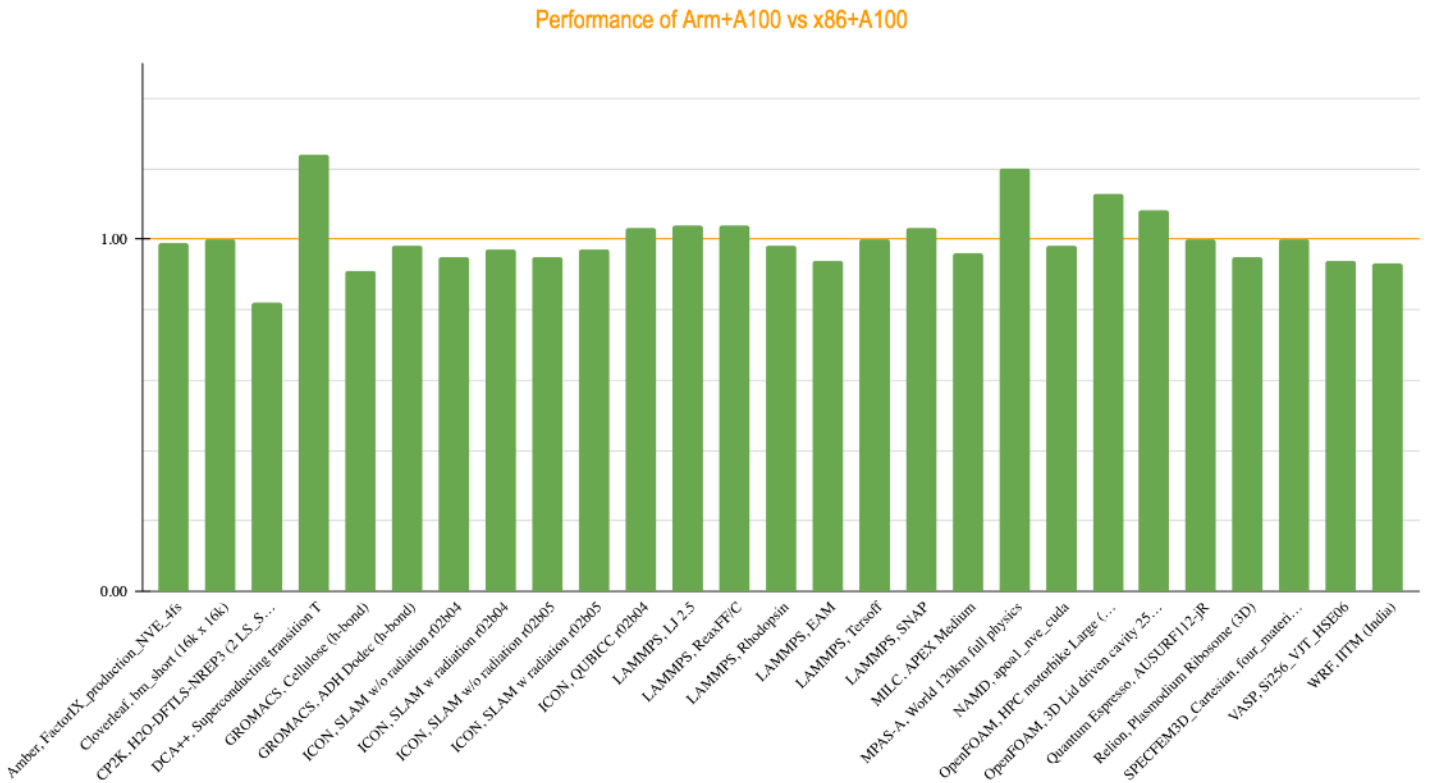
NVIDIA ARM HPC SW ECOSYSTEM



BROAD RANGE OF HPC APPLICATIONS PERFORMANT ON ARM + GPU TODAY

Performance Comparison Arm+A100 vs x86+A100

- Toolchains: NVIDIA HPC SDK 21.2, Arm Compiler for Linux 20.3, GCC 10.2
- Applications:
  - Quantum Chemistry: CP2K, Quantum Espresso, VASP
  - Molecular Dynamics: Amber, GROMACS, LAMMPS, NAMD
  - Climate/Weather: HYCOM, ICON, MPAS-A, NEMO, WRF
  - Physics: DCA++, MILC
  - Fluid Dynamics: Cloverleaf, OpenFOAM
  - Imaging: RELION
  - Seismic: SPECFEM3D

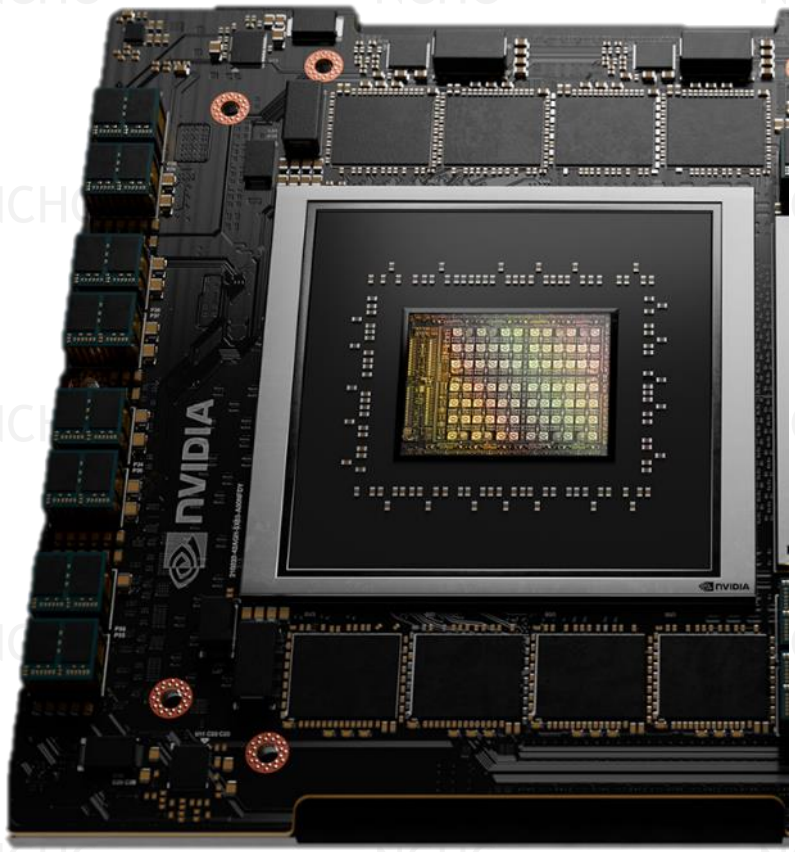


NVIDIA CONFIDENTIAL. DO NOT DISTRIBUTE.


GRACE ARM CPU

NVIDIA GRACE

Breakthrough CPU Designed for Giant Scale AI and HPC Applications

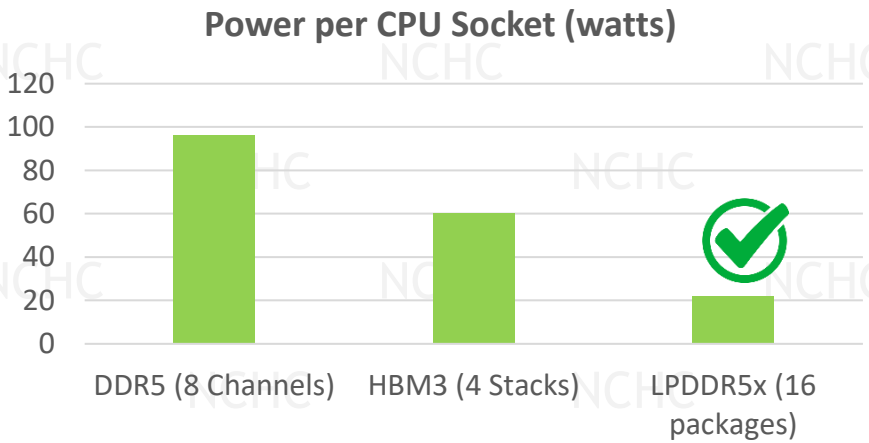


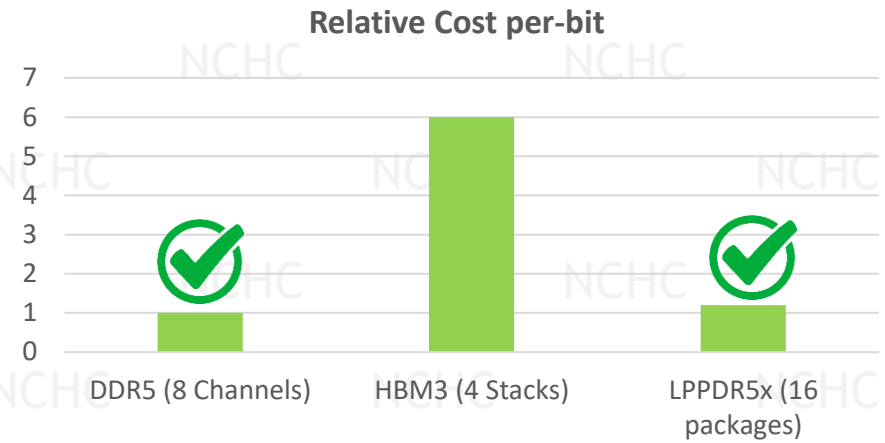
Technology	Capability
Up to 72 Cores @ >3GHz	Server Class Arm v9, SPECrate2017_int >300 >4x More Cores per GPU Allows 8 Cores per MIG + 8 for OS/system
LPDDR5	~500 GB/s Raw Bandwidth, ~400 GB/s Stream Triad, RAS 4x lower power than DDR5 Up to 480GB per Socket Arm v9 for high delivered memory performance
TDP	Up to 250W including LPDDR5 power Dynamic power sloshing between CPU and GPU
CPU to CPU NVLINK	Dual socket node: 900GB/s bi-directional coherent link 4 socket node: 600GB/s bi-directional coherent link per CPU. 2TB/s Memory Bandwidth, up to 2TB Per Node
PCIE	Up to 68 Lanes PCIE-Gen5/socket Up to 4 x16 or 8 x8 plus 2 x2 for management
CPU to GPU NVLINK	900GB/s bi-direction CPU-GPU Coherent Link

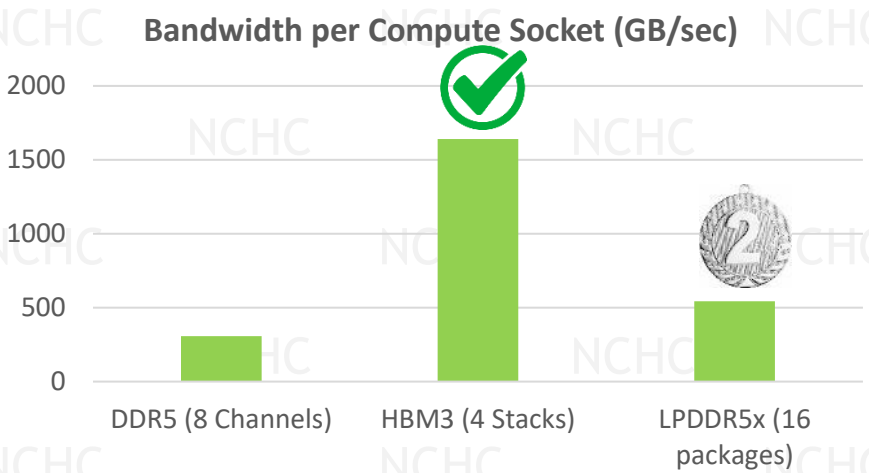
Preliminary disclosure, subject to change
Max chip capability shown, product SKUs specs can be different

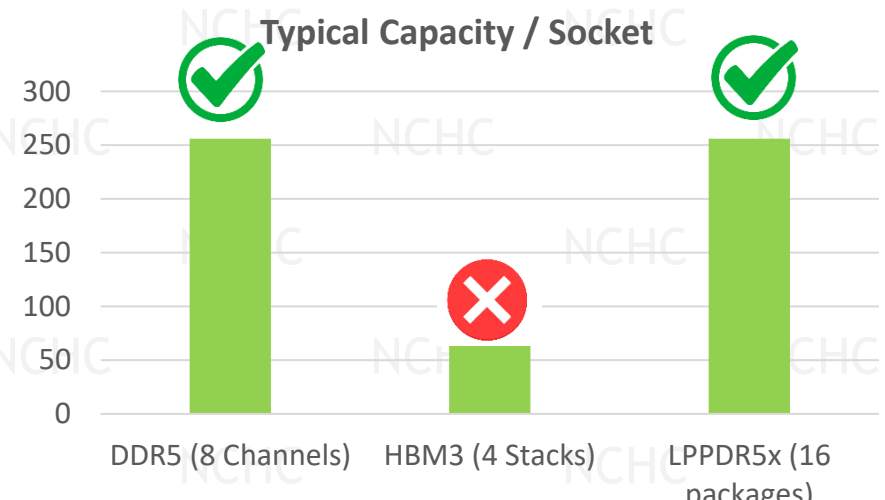
WHY WE LEVERAGE LPDDR5X

Power, Bandwidth, Capacity and Cost





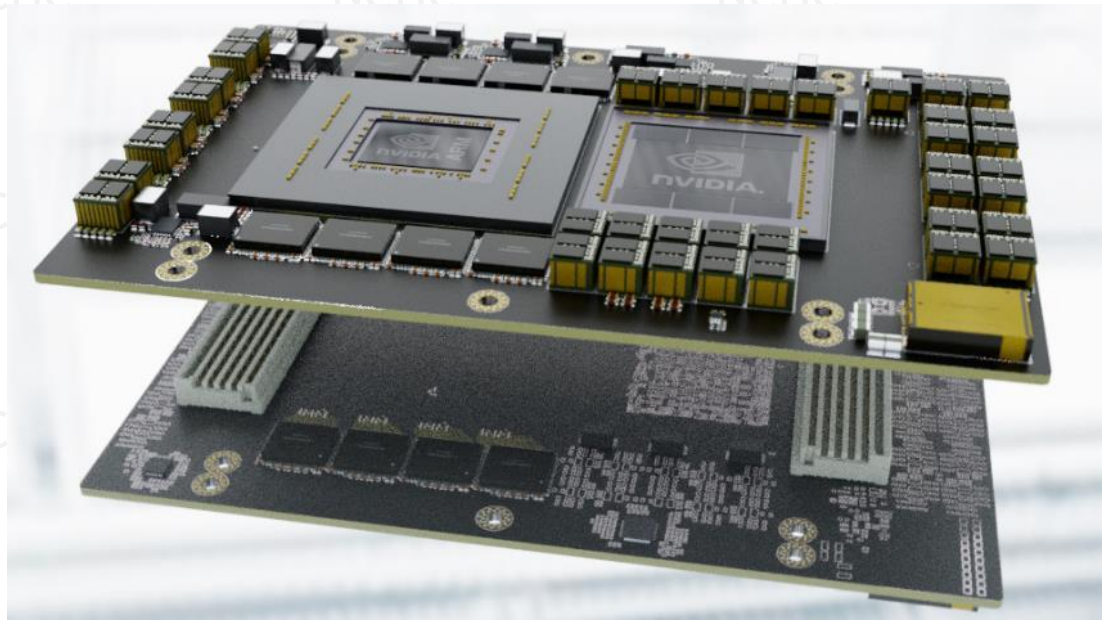




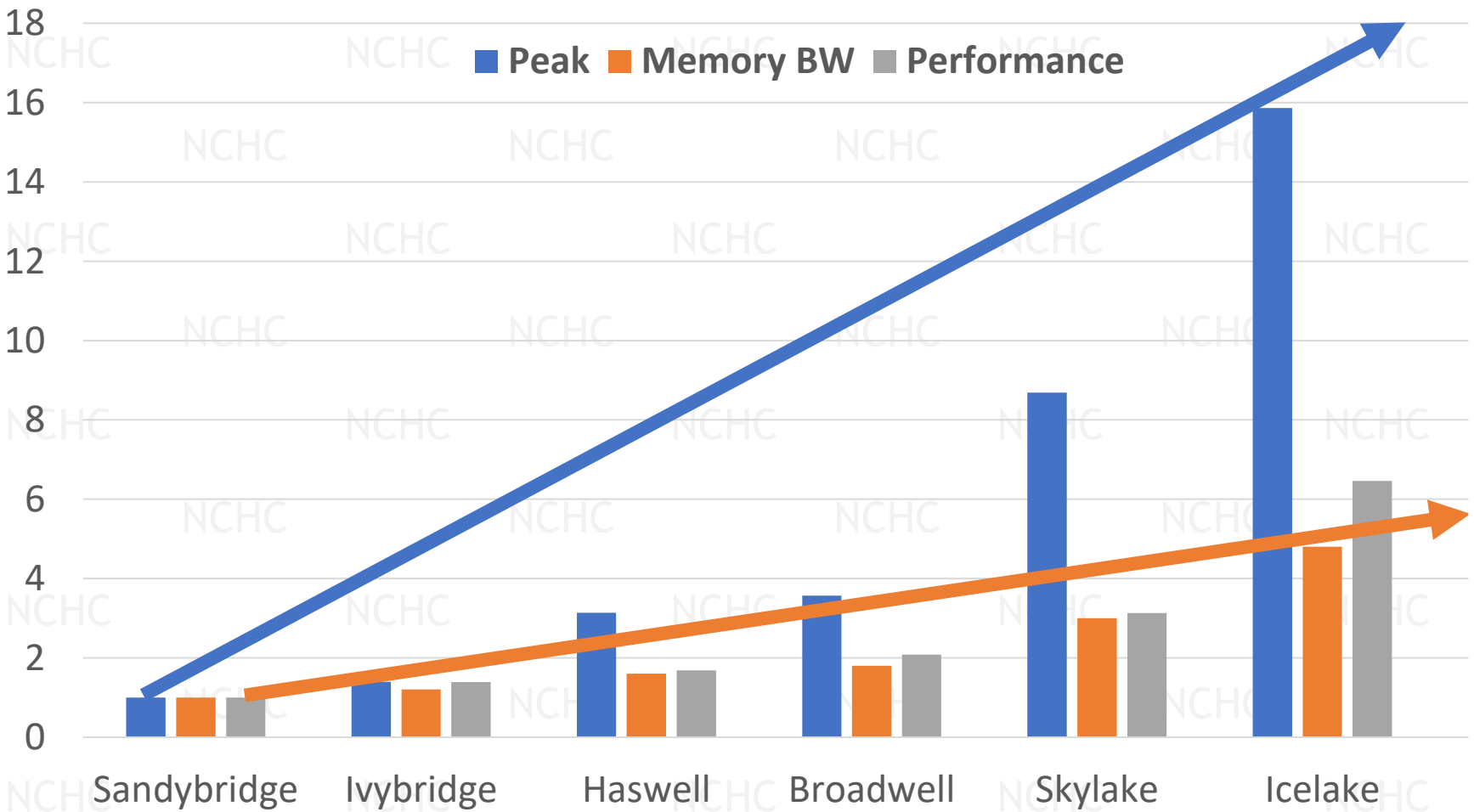
RESILIENCY AND SERVICE MODEL

LPDDR5

- Resiliency
  - Grace memory will support single-bit error correction and multi-bit error detection
- Memory Service Model
  - A heal-in-place strategy is employed
  - Spare channels are available to map out a bad memory die
  - Solid single-bit errors or uncorrectable errors can be healed by retiring the offending page from service reducing capacity by 4k
  - Expected service life of the part is ~10 years



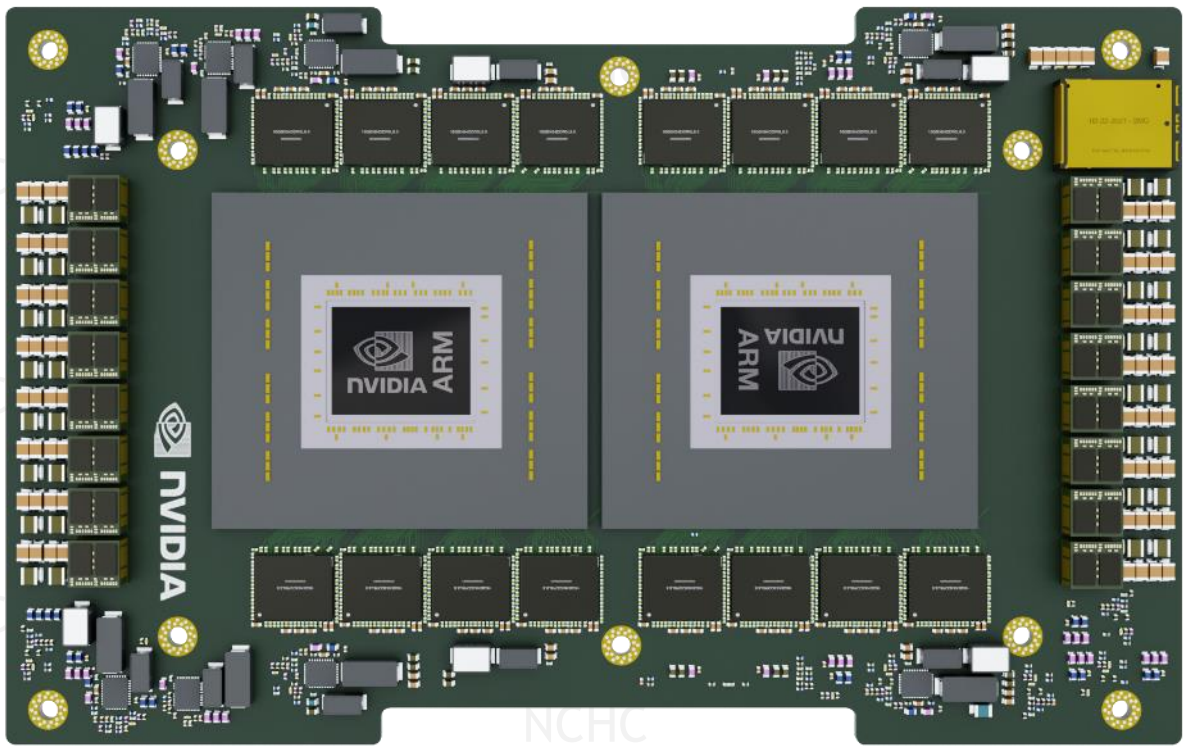
WHY ARE WE BULLISH ON GRACE FOR HPC?

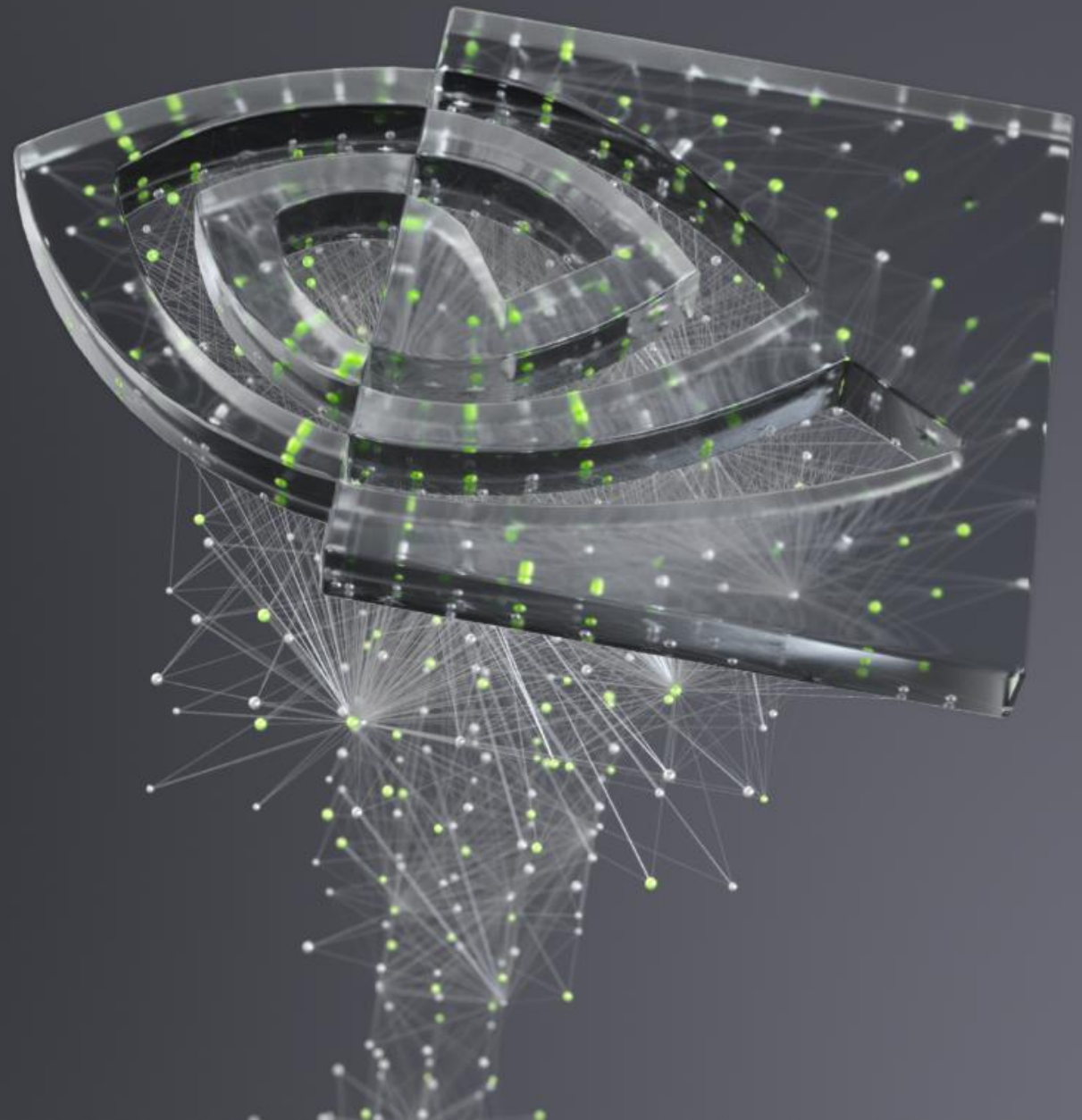


NVIDIA GRACE ONLY MODULE

Dual Grace CPU

- Fast chip-to-chip interface between processors
- Up to 144 Cores
- Up to ~1TB LPDDR5x
- 1TB/sec memory Bandwidth
- ~500 Watts
- Enables twice the packaging density of DIMM-based solutions





PROGRAMMING THE PLATFORM

PROGRAMMING THE NVIDIA PLATFORM

Develop However and Wherever You Want

	Parallel API	Incremental Directives	Standard Languages
On PCs 	NVIDIA CUDA	OpenACC OpenMP	C++ F python
On Prem 			
At the Edge			
In the Public Cloud			

Acceleration Libraries
(AI, Data Analytics, Math, Core, Quantum, Communication)

```
static NCHCine
void CalcHydroConstraintForElems(Domain &domain, Index_t length,
                 Index_t *regElemlist, Real_t dvovmax, Real_t& dthydro)
{
#if _OPENMP
  const Index_t threads = omp_get_max_threads();
  Index_t hydro_elem_per_thread[threads];
  Real_t dthydro_per_thread[threads];
#else
  Index_t threads = 1;
  Index_t hydro_elem_per_thread[1];
  Real_t dthydro_per_thread[1];
#endif
#pragma omp parallel firstprivate(length, dvovmax)
  {
    Real_t dthydro_tmp = dthydro ;
    Index_t hydro_elem = -1 ;
#if _OPENMP
    Index_t thread_num = omp_get_thread_num();
#else
    Index_t thread_num = 0;
#endif
#pragma omp for
    for (Index_t i = 0 ; i < length ; ++i) {
      Index_t indx = regElemlist[i] ;

      if (domain.vdov(indx) != Real_t(0.)) {
        Real_t dtdvov = dvovmax / (FABS(domain.vdov(indx))+Real_t(1.e-20)) ;

        if ( dthydro_tmp > dtdvov ) {
          dthydro_tmp = dtdvov ;
          hydro_elem = indx ;
        }
      }
    }
    dthydro_per_thread[thread_num] = dthydro_tmp ;
    hydro_elem_per_thread[thread_num] = hydro_elem ;
  }
  for (Index_t i = 1; i < threads; ++i) {
    if(dthydro_per_thread[i] < dthydro_per_thread[0]) {
      dthydro_per_thread[0] = dthydro_per_thread[i];
      hydro_elem_per_thread[0] = hydro_elem_per_thread[i];
    }
  }
  if (hydro_elem_per_thread[0] != -1) {
    dthydro = dthydro_per_thread[0] ;
  }
  return ;
}
```

C++ with OpenMP

STANDARD C++

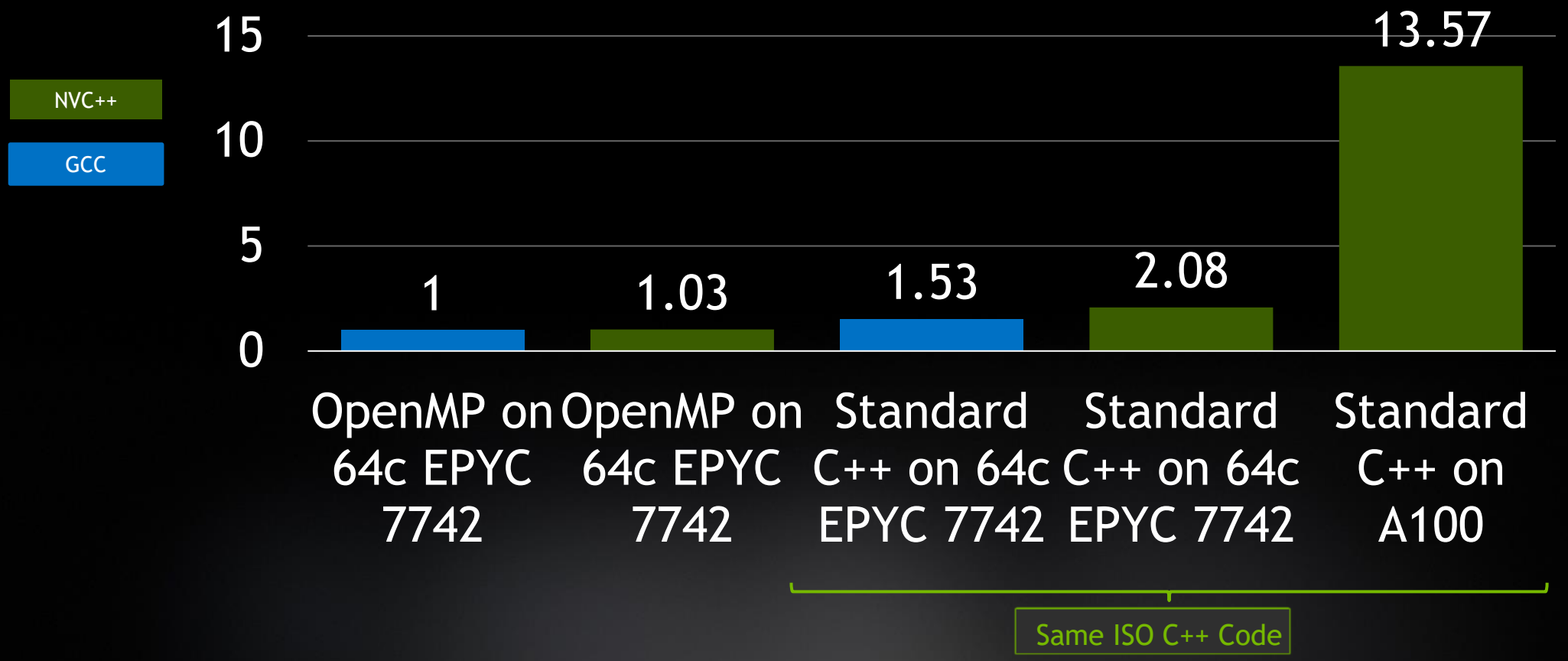
- Composable, compact and elegant
- Easy to read and maintain
- ISO Standard
- Portable – nvc++, g++, icpc, MSVC, …

```
static NCHCine void CalcHydroConstraintForElems(Domain &domain, Index_t length,
                                                Index_t *regElemlist,
                                                Real_t dvovmax,
                                                Real_t &dthydro)
{
  dthydro = std::transform_reduce(
    std::execution::par, counting_iterator(0), counting_iterator(length),
    dthydro, [](Real_t a, Real_t b) { return a < b ? a : b; },
    [=, &domain](Index_t i)
  {
      Index_t indx = regElemlist[i];
      if (domain.vdov(indx) == Real_t(0.0)) {
        return std::numeric_limits<Real_t>::max();
      } else {
        return dvovmax / (std::abs(domain.vdov(indx)) + Real_t(1.e-20));
      }
  });
}
```

Standard C++

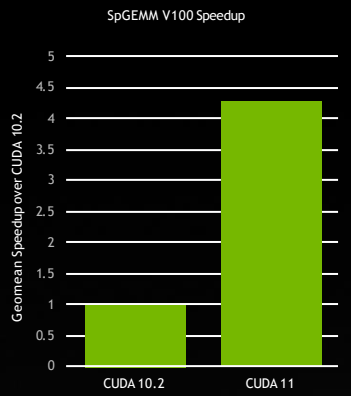
C++ STANDARD PARALLELISM

Lulesh Performance

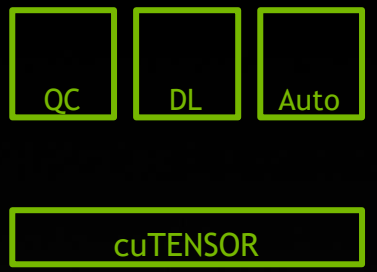


NVIDIA MATH **LIBRARIES**

Major Initiatives



Performance
Tuning
New algorithms



Extended Features
New libraries
New APIs

Multi-GPU
Strong Scaling
Weak scaling

Single GPU
Tensor Cores
Device Functions

HPC COMPILERS

NVC | NVC++ | NVFORTRAN

Accelerated

A100
Automatic

Fortran

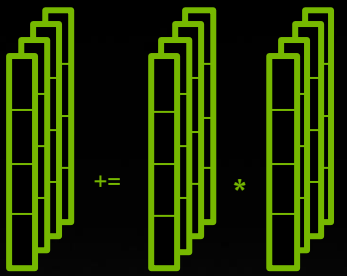
NVIDIA CUDA

OpenACC
More Science, Less Programming

OpenMP

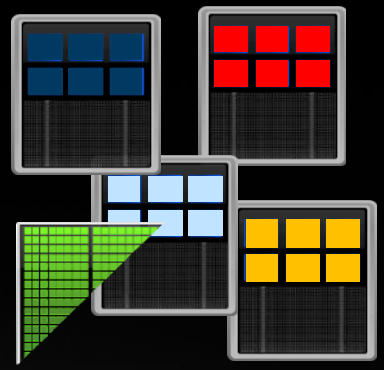
Programmable

Standard Languages
Directives
CUDA



Multicore

Directives
Vectorization



Multi-Platform

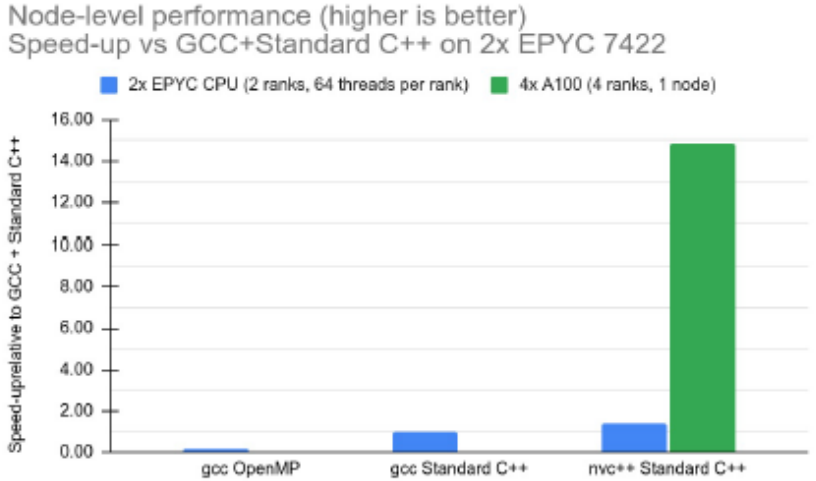
x86_64
Arm
OpenPOWER

NVIDIA

MAIA

Ported to Standard C++17 Parallelism

MAIA (ZFS): Node-level performance
2x EPYC 7422 + 4x A100



Implementation of a Lattice Boltzmann method for the analysis of landing-gear noise on GPU based HPC Systems

M. Waldmann[1], G. Brito-Gadeschi[3], M. Gondrum[1], M. Meinke[1,2], and W. Schröder[1,2]

[1]Chair of Fluid Mechanics and Institute of Aerodynamics, RWTH Aachen University, Wüllnerstraße 5a, 52062 Aachen, Germany
[2]Center for Simulation and Data Science, RWTH Aachen University, Kopernikusstraße 6, 52074 Aachen, Germany
[3]NVIDIA GmbH, Adenauerstraße 20 A4, 52146 Würselen, Germany
✉Corresponding author: Moritz Waldmann, m.waldmann@aia.rwth-aachen.de

August 15, 2021

...nt of aircraft noise during approach is generated by airframe components such as ...l the landing-gear (LG). To mitigate the aerodynamically generated sound from such ...arly stage of the design process, it is necessary to accurately predict the acoustic ...he flow field around an LG configuration is predicted as a first step by large eddy ...nalyze the various noise sources from the LG's sub-components, i.e., torque links and ...t a deeper understanding of the noise generation mechanisms, which is necessary ...noise mitigation techniques.
...n LES require large computational resources, an implementation on low cost HPC ...us. In particular, graphical processing units (GPUs) have an attractive price to ...h is the reason why such hardware has become popular in the recent years. A ...is the necessity to port the simulation software for the execution on GPUs. In ...nming models such as CUDA, OpenCL, or OpenACC were developed, which ...alls or directives into the program code. These differ, among others, in their ...PU processing units of various vendors and CPU-based systems. Since many HPC hardware ...t suitable for compilers of CPU-based systems. The CUDA ...AWK system installed at HLRS Stuttgart, are still purely CPU-based systems, ...lation framework applied to investigate diverse physical phenomena can be ...stems, i.e., CPU and GPUs. This could be achieved by inserting directives, ...owever, ... and off the various software section for the CPU and GPU implementation. This, ...introduces code duplication and thus substantial overhead for software maintenance. Therefore, a different strategy was pursued, by using the C++ 17 standard for the GPU porting of the lattice Boltzmann method (LBM) of the multi-physics simulation framework m-AIA, which is actively developed at the Institute of Aerodynamics, and which was formerly known as Zonal ... The C++ 17 standard is implemented in the latest Nvidia compiler ... Clang, Intel, and Gnu compiler versions, which are usu...

STLBM

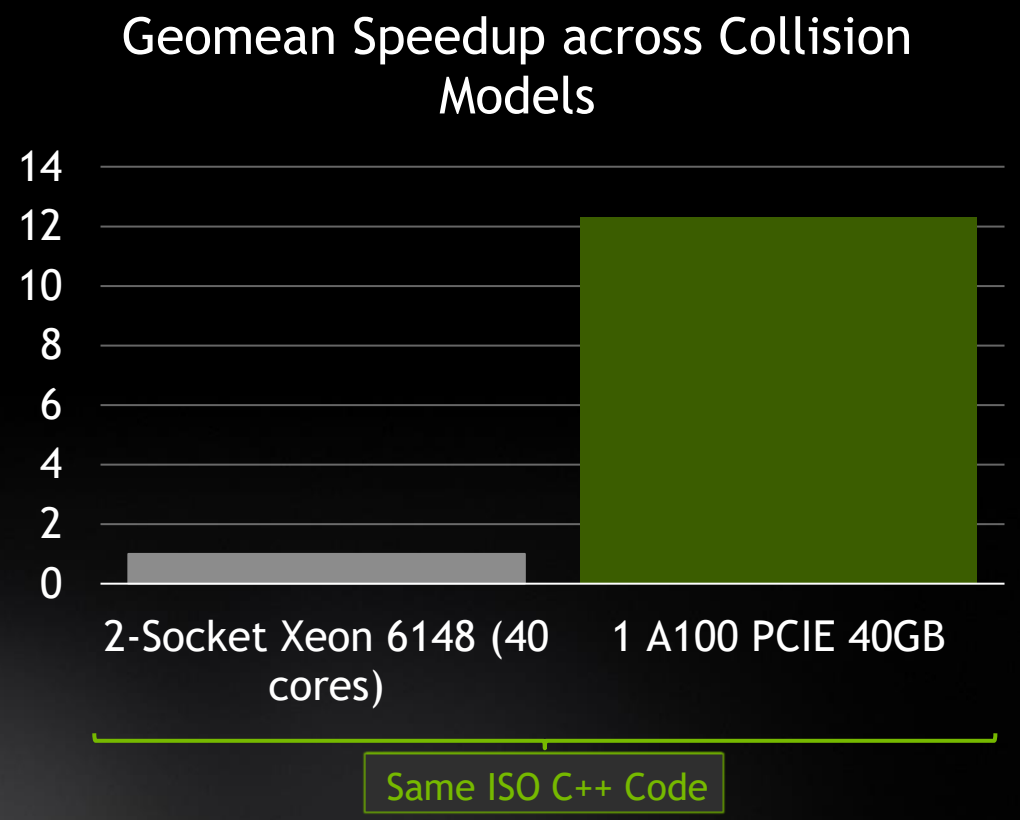
Many-core Lattice Boltzmann with C++ Parallel Algorithms

- Framework for parallel lattice-Boltzmann simulations on multiple platforms, including many-core CPUs and GPUs
- Implemented with C++17 standard (Parallel Algorithms) to achieve parallel efficiency
- No language extensions, external libraries, vendor-specific code annotations, or pre-compilation steps

"We have with delight discovered the NVIDIA "stdpar" implementation of C++ Parallel Algorithms. … We believe that the result produces state-of-the-art performance, is highly didactical, and introduces ***a paradigm shift in cross-platform CPU/GPU programming*** *in the community."*

-- Professor Jonas Latt, University of Geneva

https://gitlab.com/unigehpfs/stlbm
https://www.nvidia.com/en-us/on-demand/session/gtcspring21-s32076/



```
subroutine ref_sd_t_d1_4(h3d, h2d, h1d, p6d, p5d, p4d, &
                         h7d, t3, t2, v2)
  Implicit none
  integer h3d, h2d, h1d, p6d, p5d, p4d, h7d
  integer h3, h2, h1, p6, p5, p4, h7
  double precision t3(h3d, h2d, h1d, p5d, p4d, p6d)
  double precision t2(h7d, p4d, p5d, h1d)
  double precision v2(h3d, h2d, p6d, h7d)
  do p6 = 1, p6d
   do p4 = 1, p4d
    do p5 = 1, p5d
     do h1 = 1, h1d
      do h2 = 1, h2d
       do h3 = 1, h3d
        do h7 = 1, h7d
         t3(h3,h2,h1,p5,p4,p6) = t3(h3,h2,h1,p5,p4,p6) &
                               - t2(h7,p4,p5,h1) * &
                                 v2(h3,h2,p6,h7)
        enddo
       enddo
      enddo
     enddo
    enddo
   enddo
  enddo
end
```

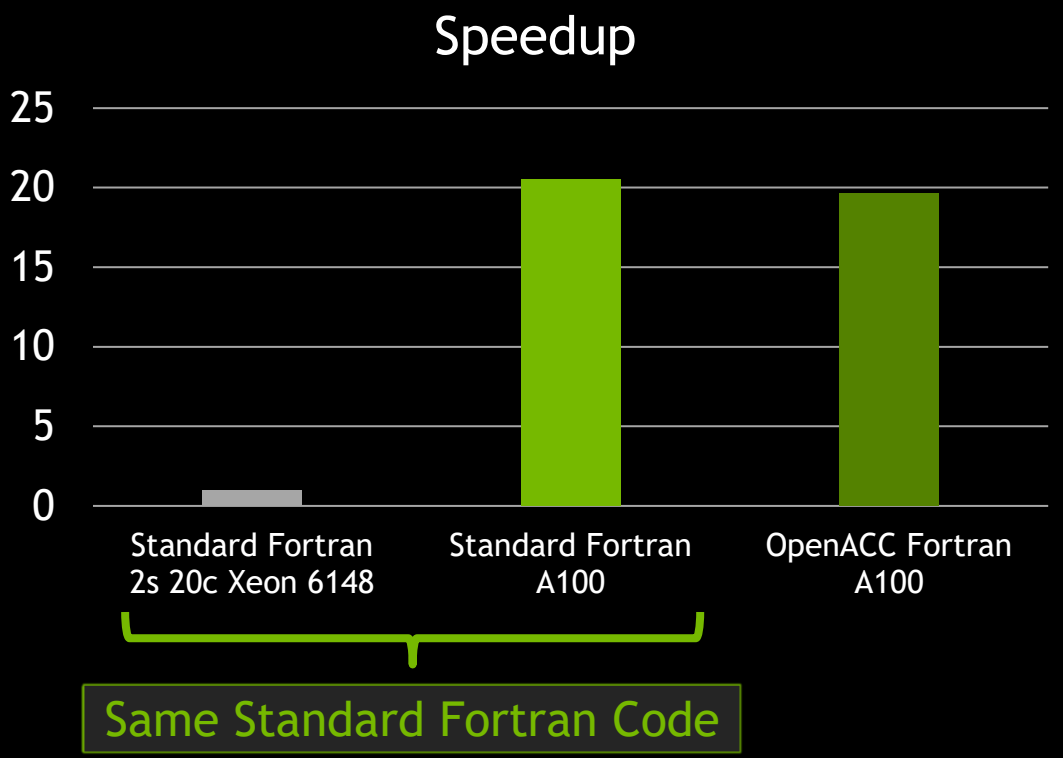
Standard Fortran (Serial)

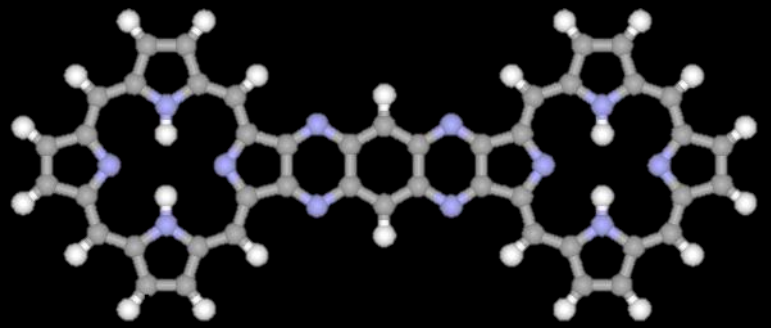
```
subroutine par_sd_t_d1_4(h3d, h2d, h1d, p6d, p5d, p4d, &
                         h7d, t3, t2, v2)
  Implicit none
  integer h3d, h2d, h1d, p6d, p5d, p4d, h7d
  integer h3, h2, h1, p6, p5, p4, h7
  double precision t3(h3d, h2d, h1d, p5d, p4d, p6d)
  double precision t2(h7d, p4d, p5d, h1d)
  double precision v2(h3d, h2d, p6d, h7d)
  do concurrent (p6 = 1 : p6d)
   do concurrent (p4 = 1 : p4d)
    do concurrent (p5 = 1 : p5d)
     do concurrent (h1 = 1 : h1d)
      do concurrent (h2 = 1 : h2d)
       do concurrent (h3 = 1 : h3d)
        do h7=1,h7d
         t3(h3,h2,h1,p5,p4,p6) = t3(h3,h2,h1,p5,p4,p6) &
                               - t2(h7,p4,p5,h1) *
                                 v2(h3,h2,p6,h7)
        enddo
       enddo
      enddo
     enddo
    enddo
   enddo
  enddo
end
```

Standard Fortran (Parallel for CPUs and GPUs)

NWChem TCE CCSD(T) Kernel

Computational Chemistry with Fortran Standard Parallelism





- NWChem provides a massively parallel implementation of the "gold standard" CCSD(T) method that scales to hundreds of thousands of CPU cores.
- The compute bottleneck is a set of 27 loop-driven tensor contractions, which are part of the >100k LOC TCE module.

https://github.com/jeffhammond/nwchem-tce-triples-kernels

NVIDIA


APPLICATION
PERFORMANCE

WHERE IS GRACE+HOPPER BETTER THAN X86+HOPPER?

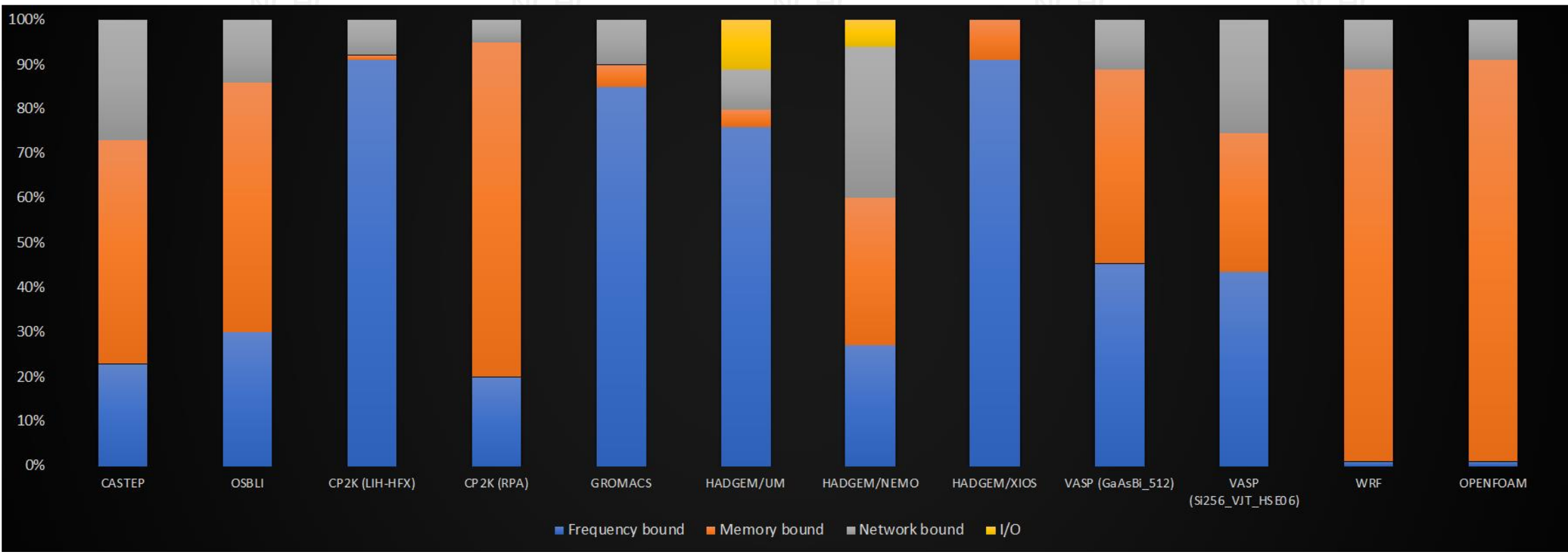
Some cases

Partially Ported Apps	Apps that bottleneck on PCI connectivity	Apps that can leverage tight cache coherence	New-to-GPU Apps
• OpenFOAM – solver only (bar is lower to better price/perf)	• ABINIT example with pencil-shaped ZGEMM • Large AI Training	• Data Assimilation step in weather models can stay on Grace	• Can more effectively leverage standard language acceleration

APPLICATION PERFORMANCE METHODOLOGY

CHARACTERIZATION OF KEY APPLICATIONS

- Simulation of GROMACS and WRF including hot-spot analysis
- Estimation model for all the other applications



NVIDIA GRACE CPU: PERFORMANT AND POWER EFFICIENT

Relative Performance

Application	AMD Rome 7742 *(Dual Socket)*	NVIDIA GRACE *(Dual Socket)*
CASTEP	1	1.6
GROMACS*	1	1.5
VASP (GaAsBi_512)	1	1.8
WRF*	1	2.2
OPENFOAM	1	2.1
CP2K (RPA)	1	2.1
CP2K (LIH-HFX)	1	1.5
OSBLI	1	1.8
HADGEM3 (average)	1	1.5
Power (CPU+MEM)	1	0.8

Performance is estimates, except for * where performance is based on preliminary system simulation
GRACE dual socket performance assumes a high core count NVIDIA GRACE CPU

Preliminary disclosure, subject to change. Max chip capability shown, product SKUs specs can be different

NVIDIA GRACE + ANEXT EVEN BETTER COMBINATION

Relative Performance

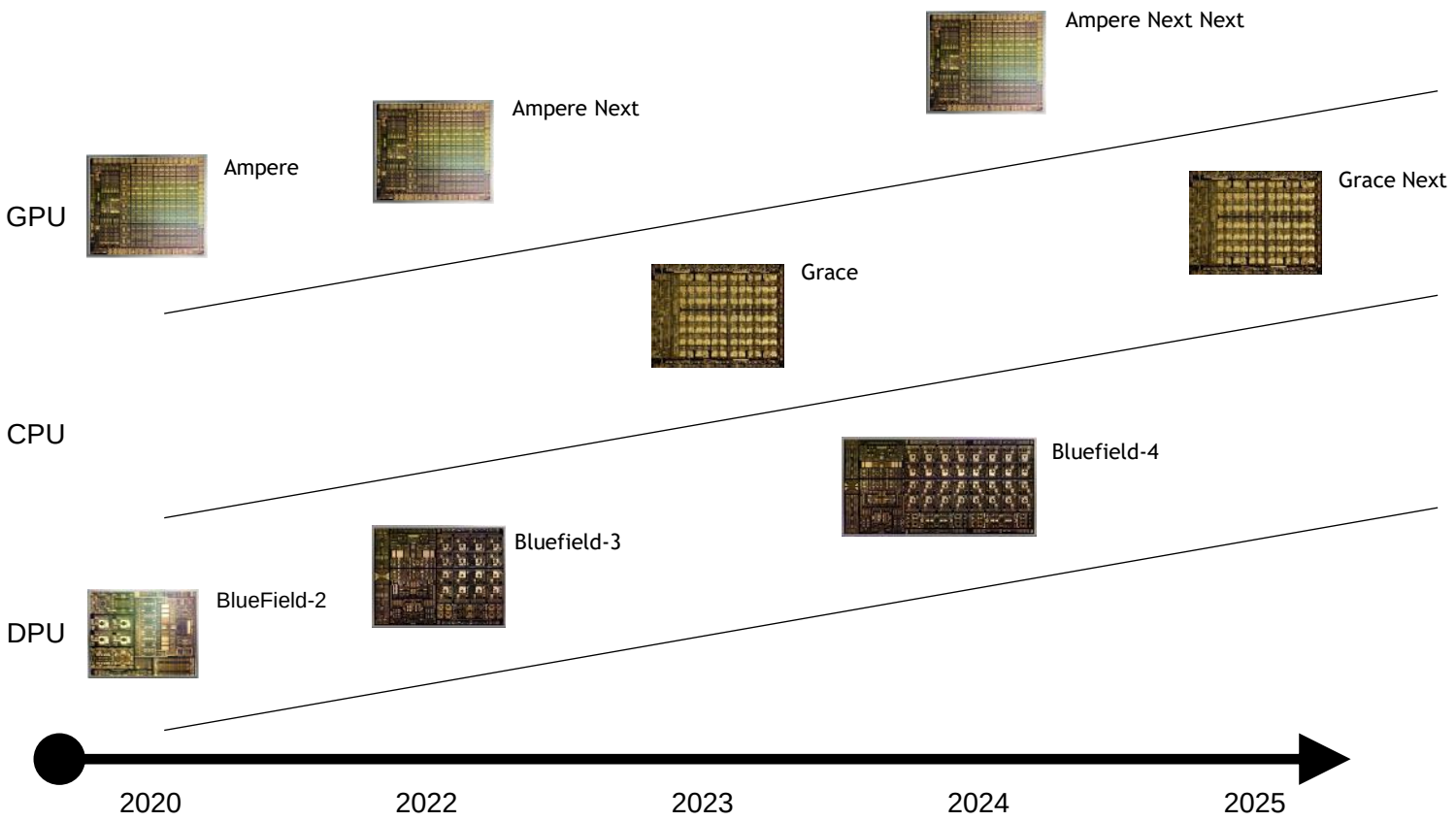
Application	AMD Rome 7742 *(Dual Socket)*	NVIDIA GRACE *(Dual Socket)*	GRACE + ANEXT *(Single Module)*
CASTEP	1	1.6	8.2**
GROMACS*	1	1.5	7.8
VASP (GaAsBi_512)	1	1.8	9.0
WRF*	1	2.2	11.3
OPENFOAM	1	2.1	3.5
CP2K (RPA)	1	2.1	8.4
CP2K (LIH-HFX)	1	1.5	0.8
OSBLI	1	1.8	1.1
HADGEM3 (average)	1	1.5	0.9
Power (CPU+MEM)	1	0.8	1.3

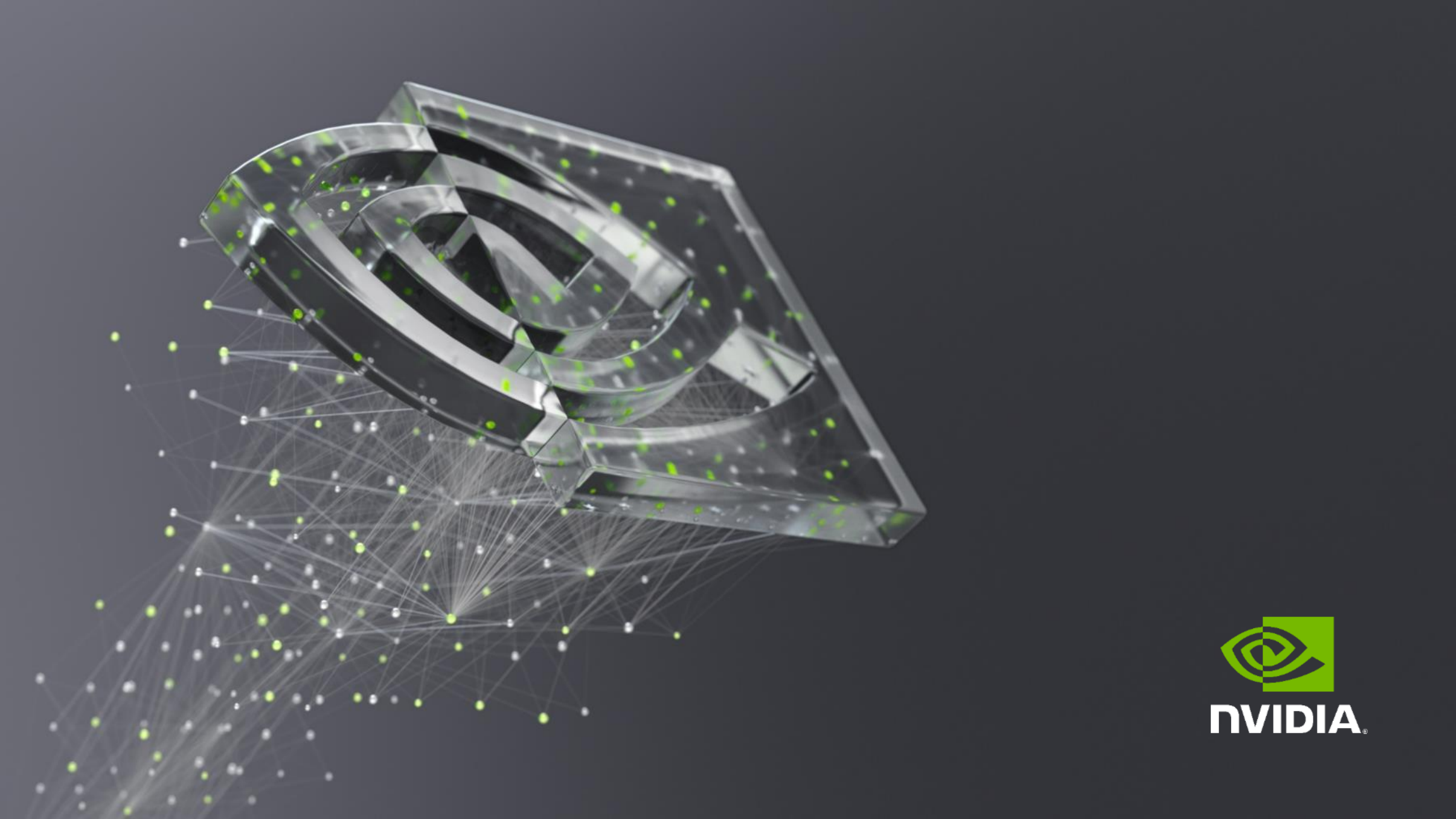
Performance is estimates, except for * where performance is based on preliminary system simulation
** CASTEP performance assumes all functionality accelerated on GPU
GRACE+ANEXT module performance assumes a medium core count NVIDIA GRACE CPU
GRACE dual socket performance assumes a high core count NVIDIA GRACE CPU

Zero GPU content assumed

Preliminary disclosure, subject to change. Max chip capability shown, product SKUs specs can be different

NVIDIA DATA CENTER ROADMAP




NVIDIA