Aurora

Intel's First Exascale System Architecture

Olivier Franza
Senior Principal Engineer, Advanced Supercomputing Architecture
Aurora Chief Architect and Principal Investigator

intel.

Notices and Disclaimers

Statements in this document that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.

All product plans and roadmaps are subject to change without notice.

Performance varies by use, configuration and other factors. Learn more on the Performance Index site. Intel technologies may require enabled hardware, software or service activation.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

intel.

Aurora Spec Sheet

Compute

21,248
CPUs

63,744
GPUs

10,624
Nodes

Fabric

Peak Injection Bandwidth
2.12 PB/s

Peak Bisection Bandwidth
0.69 PB/s

Dragonfly Topology

Memory

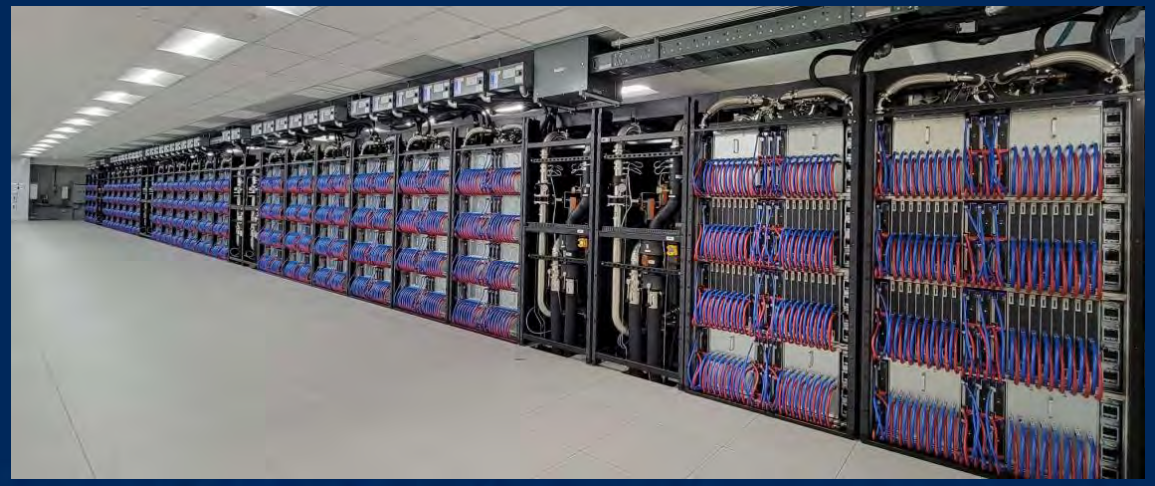
10.9PB	**1.36PB**	**8.16PB**
DDR Capacity	HBM CPU Capacity	HBM GPU Capacity
5.95PB/s	**30.5PB/s**	**208.9PB/s**
Peak DDR BW	Peak HBM BW CPU	Peak HBM BW GPU

Storage

230PB	**31TB/s**	**1024**
DAOS Capacity	DAOS Bandwidth	DAOS Node #

intel.

Aurora Fun Facts



Exascale = a billion billion (a quintillion) operations per second

Artificial Intelligence | Analytics | HPC Simulation

1 SECOND
- Time it takes Aurora to solve a math problem that would take 40 years if all the people on earth all did one calculation every 10 seconds.

600 TONS
- The weight of Aurora, which equals that of an Airbus A380.

300 MILES
- The length of optical cable used in Aurora, that's the distance between Boston and Montréal.

10,000 SQUARE FEET
- The amount of floor space for Aurora, or 4 tennis courts.

8 MINUTES
- The time it takes Aurora to store enough characters to write a stack of books that could reach the moon.

230 PB OF DAOS STORAGE
- That's the equivalent of 70 years of HD videos.

intel.

Aurora Mission Examples

Aurora Exascale Supercomputer to Advance Clean Fusion Research

Researchers seeking new approaches to contain fusion reactions for the generation of electricity stand ready to tap Aurora's full potential.

Watch the video

Researching Our Universe on Aurora Exascale

Research Scientist Jimmy Proudfoot talks about the impact Exascale supercomputing will have on his work researching our universe.

Watch the video

Neuroscience Research on Aurora Exascale

Senior Computer Scientist Nicola Ferrier explains how neuroscience research will process exabytes of data on the Aurora Exascale Supercomputer.

Watch the video

Propelling Aerospace Research on Aurora Exascale

Aerospace Professor Ken Jansen explains how engineers will create faster and more complex models and simulations on Exascale supercomputers.

Watch the video

CANDLE Taps Deep Learning to Identify Effective Cancer Treatments

CANcer Distributed Learning Environment (CANDLE) taps deep learning to explore the biology of cancer, and identify highly effective treatments.

Watch the video

Exascale Computing to Power Catalysts Research

Aurora's exascale capabilities will enable catalyst researchers to perform more high-fidelity simulations for better quantitative descriptions.

Watch the video

intel.

Argonne NATIONAL LABORATORY

Hewlett Packard Enterprise

intel.

Aurora genAI

State-of-the-art Generative AI Model for Science

Trained on

General text
Scientific texts
Scientific data
Code

Target Size

1 Trillion
Parameters

Foundations

Megatron
& DeepSpeed

Potential Applications

Systems Biology
Cancer Research
Climate Science
Cosmology
Polymer Chemistry & Materials Science

intel.

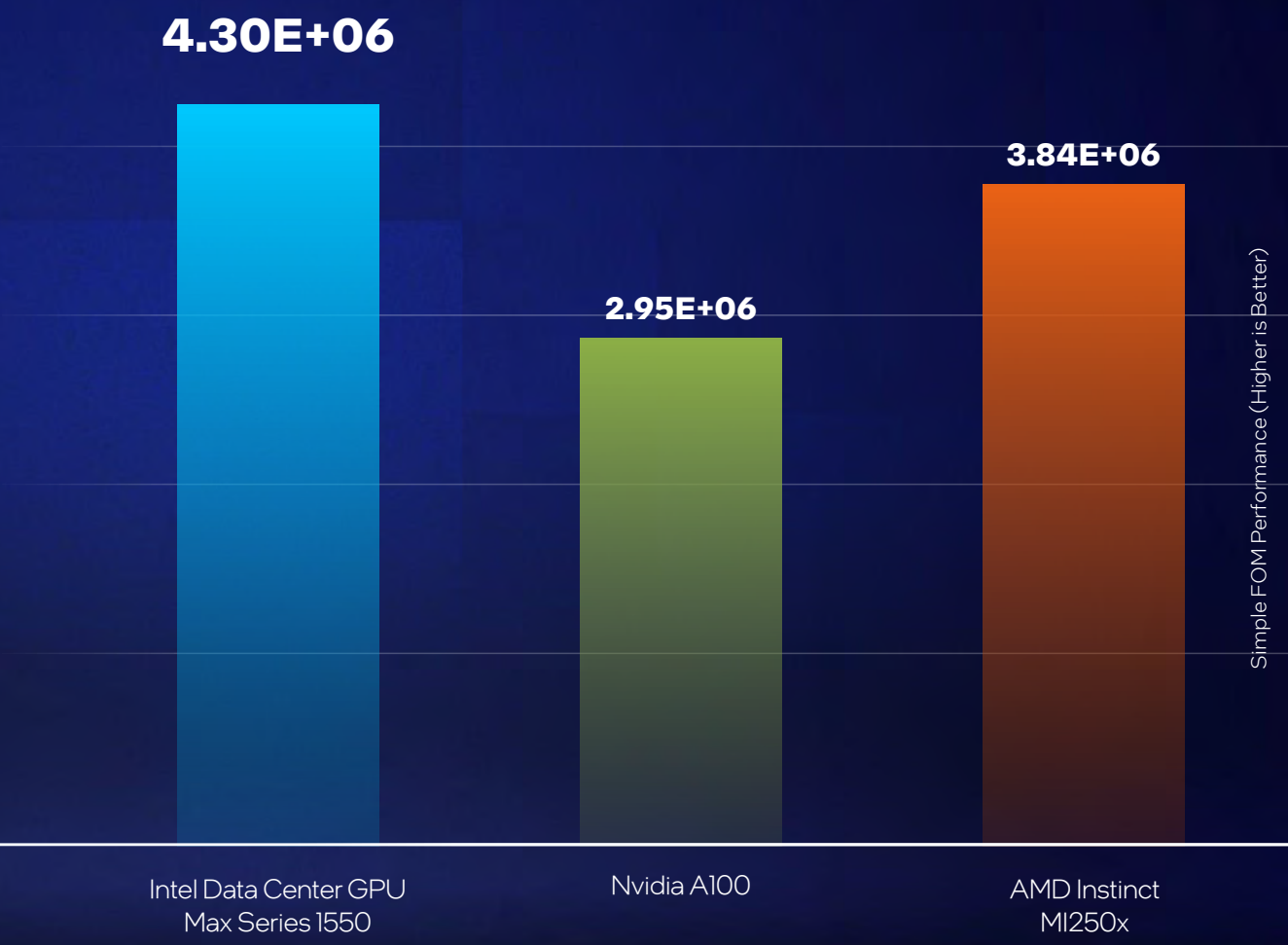
EXASCALE COMPUTING PROJECT

AURORA

Speeding up Fusion Reactor Prediction

ITER - Predicting plasma behavior with XGC

(Single-GPU Measurement)



Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy

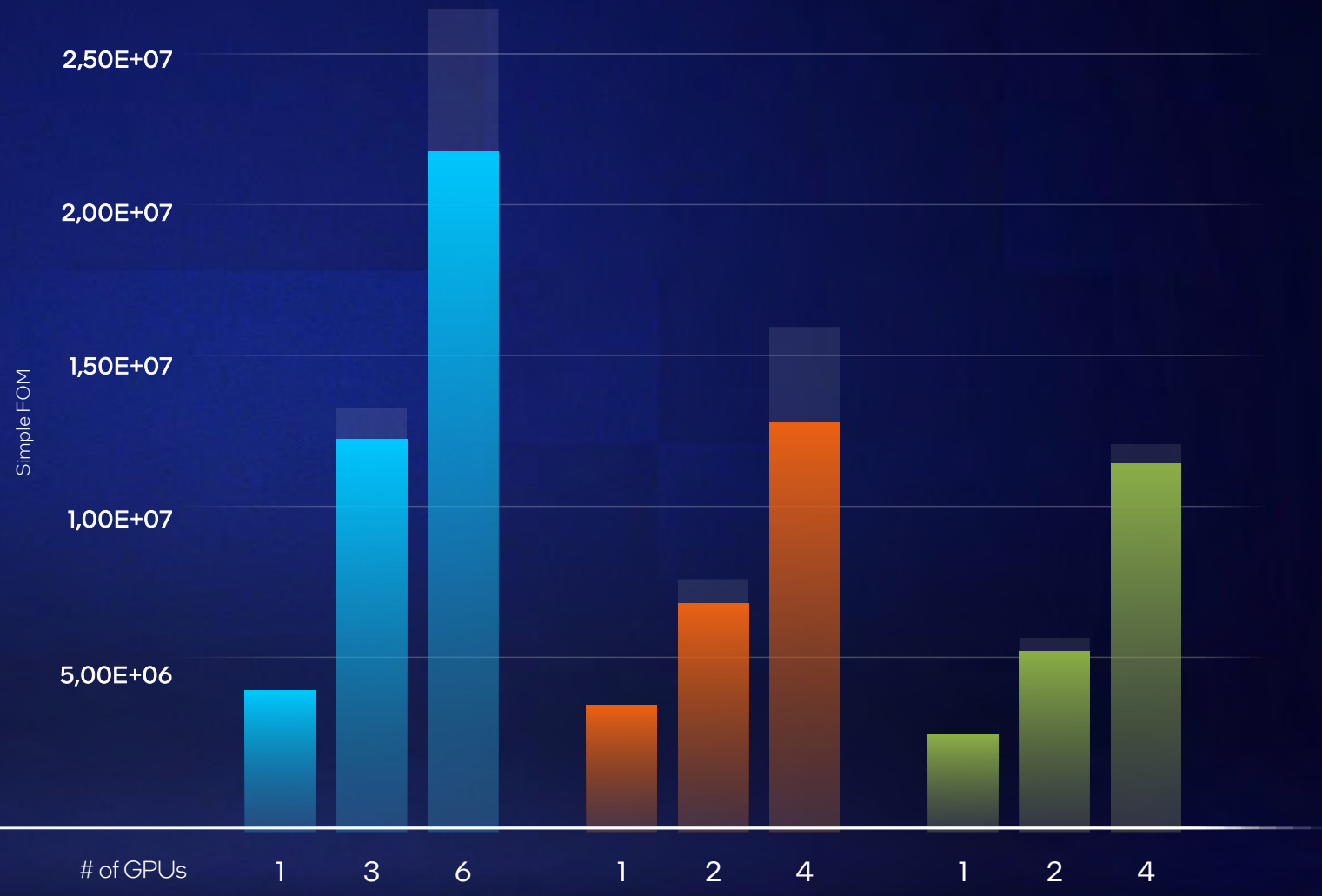
intel.

EXASCALE COMPUTING PROJECT

AURORA

Speeding up Fusion Reactor Prediction

ITER - Predicting plasma behavior with XGC



EXASCALE COMPUTING PROJECT

AURORA

Excellence at Exascale

Speeding up Fusion Reactor Prediction

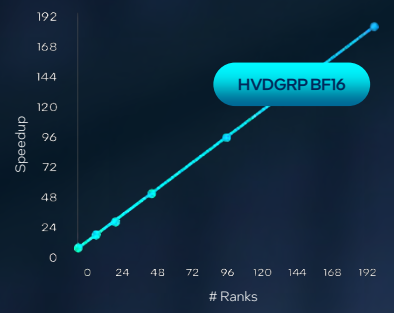
For XGC workload

1.4x

vs. Nvidia A100 PCIe

High-Energy Particle Physics at Scale

Perf Scaling on CosmicTagger



Accelerating Fusion Plasma Modelling

For WDMApp GENE

1.6x

vs. Nvidia A100 PCIe

Computing Quantum Mechanics Faster

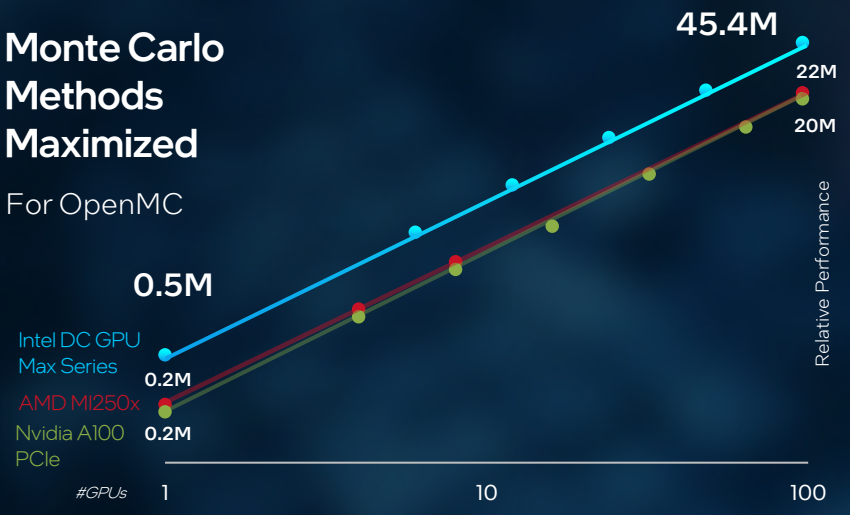
For QMCPACK

1.2x

vs. Nvidia H100 PCIe

Monte Carlo Methods Maximized

For OpenMC



Computational Chemistry at Exascale

For NWChemex

1.7x

vs. Nvidia A100

Our Brain analyzed at Exascale

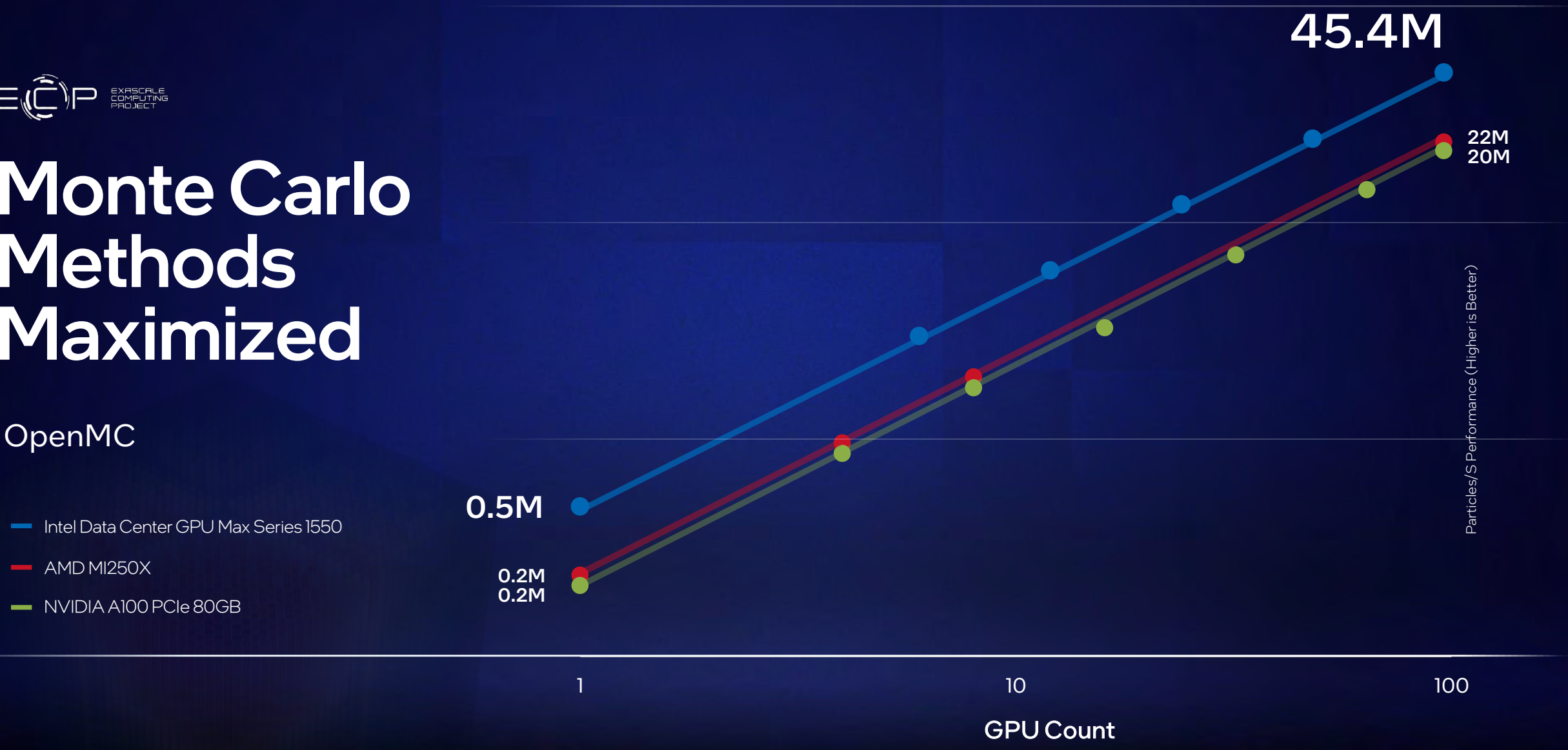
Connectomics Performance Scaling

intel.

ISC High Performance
The HPC Event

Monte Carlo Methods Maximized

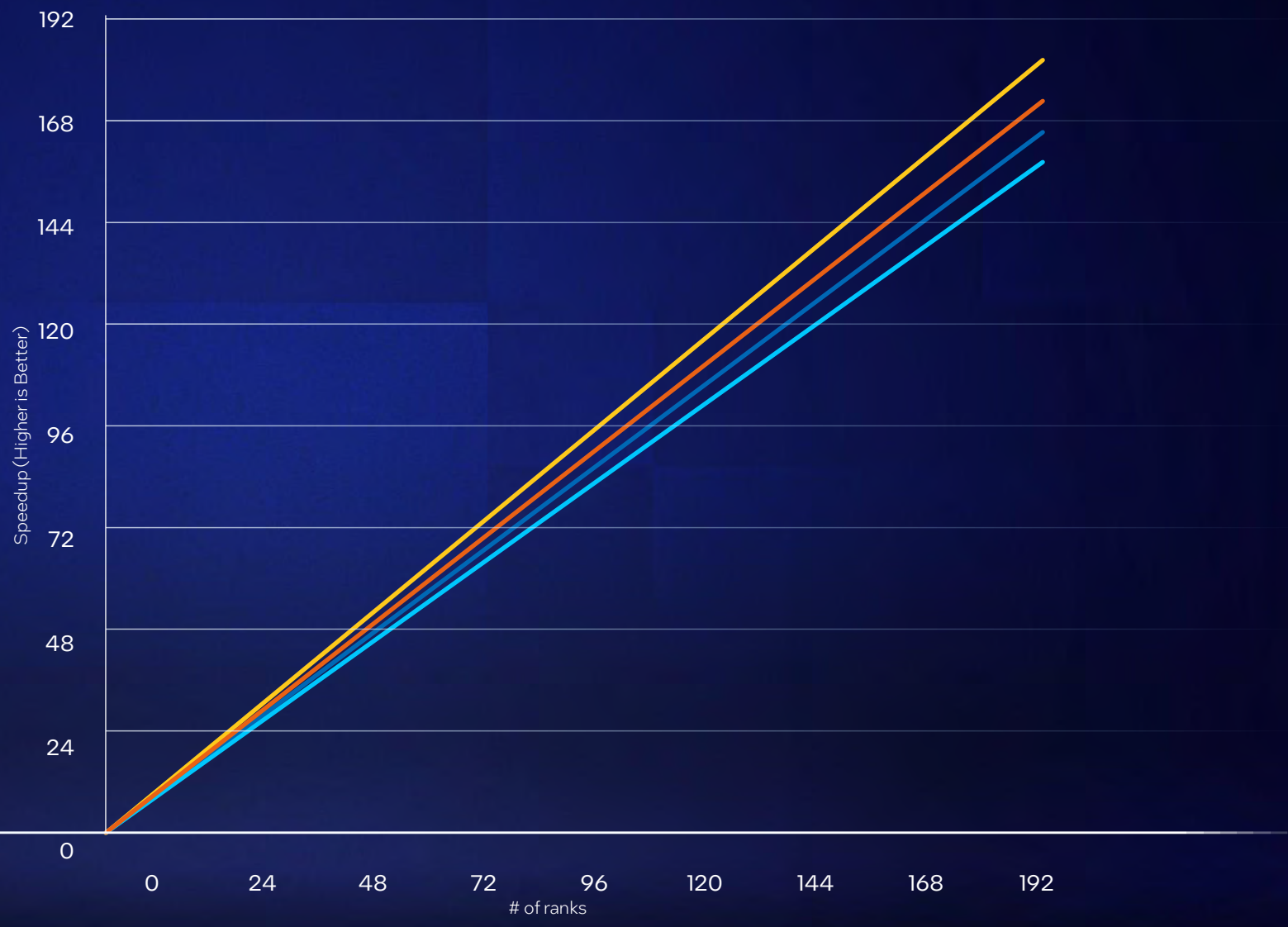
OpenMC



Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy

High-Energy Particle Physics at Scale

Scaling Cosmic Tagger

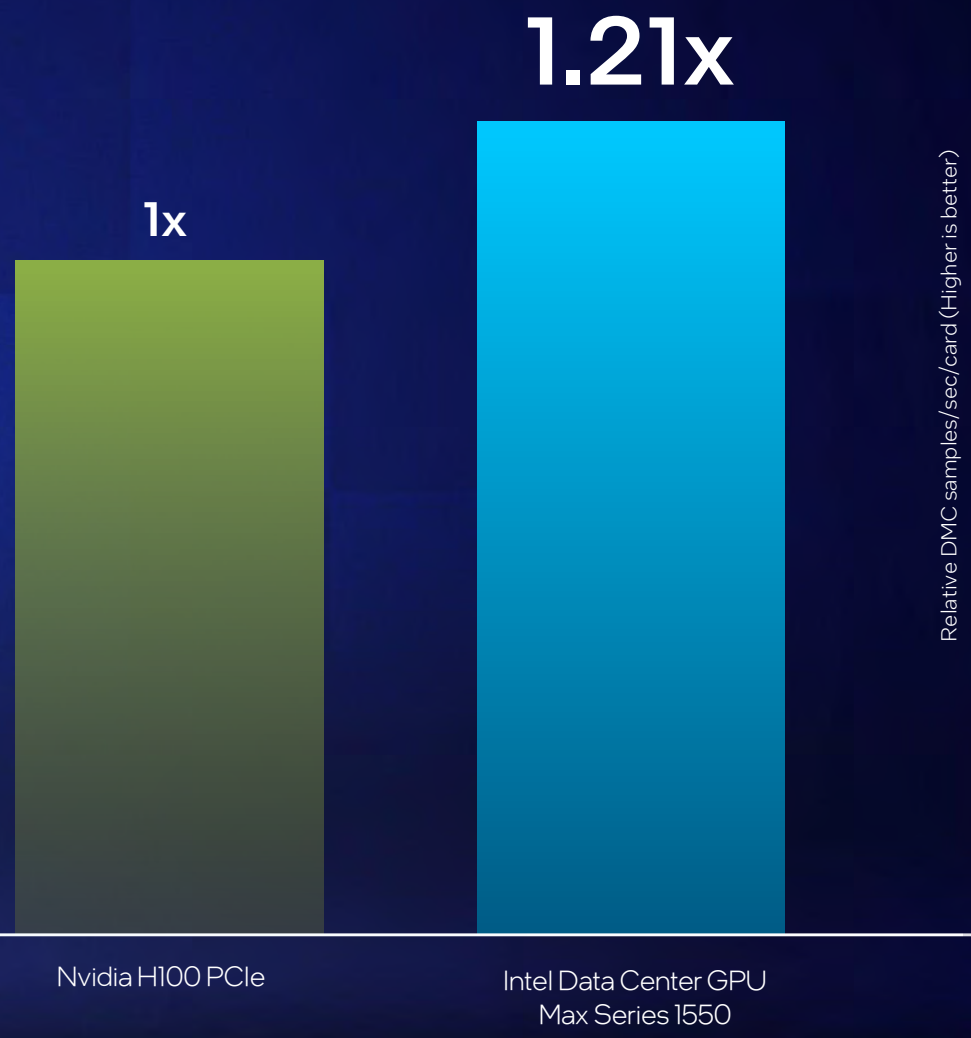


Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy

ECP EXASCALE COMPUTING PROJECT AURORA

Computing Quantum Mechanical Properties faster

QMCPACK performance
(16 walker per card)



See backup for workloads and configurations. Results may vary.

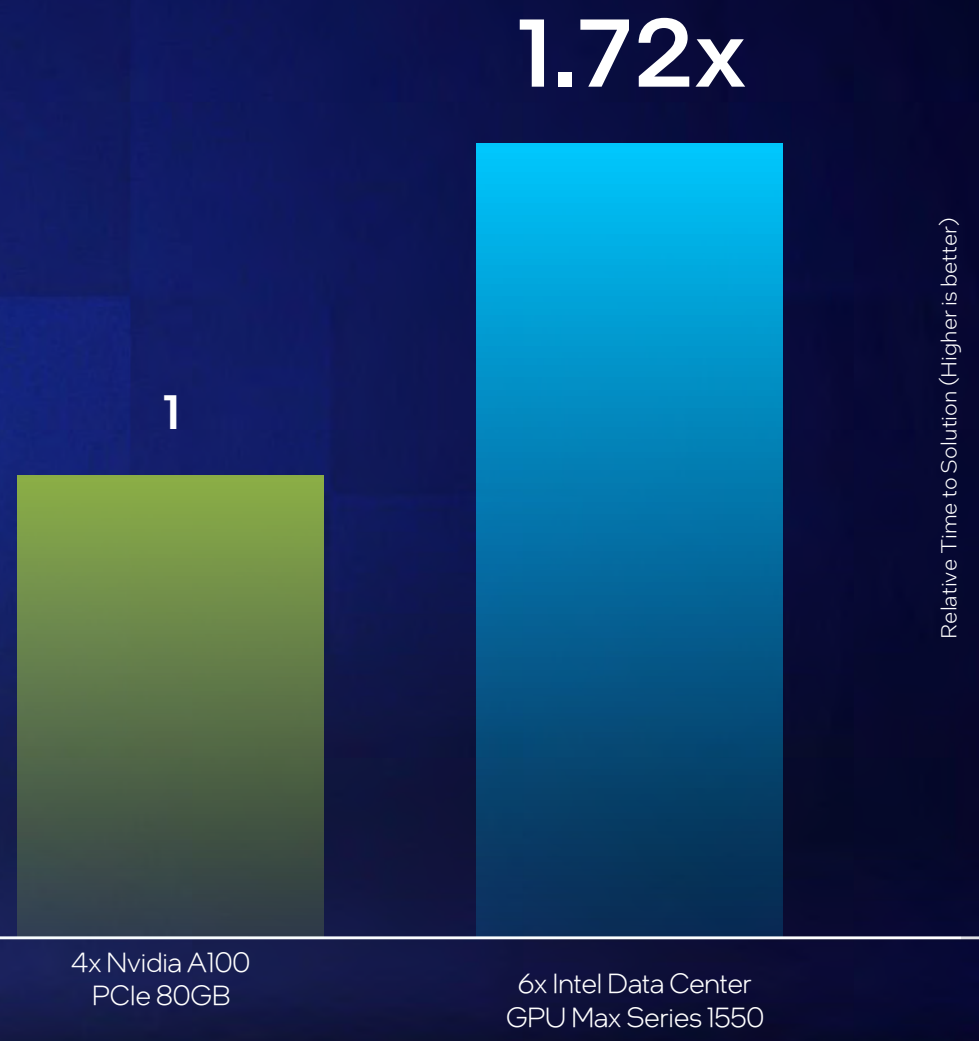
intel.

EXASCALE COMPUTING PROJECT

AURORA

Computational Chemistry at Exascale

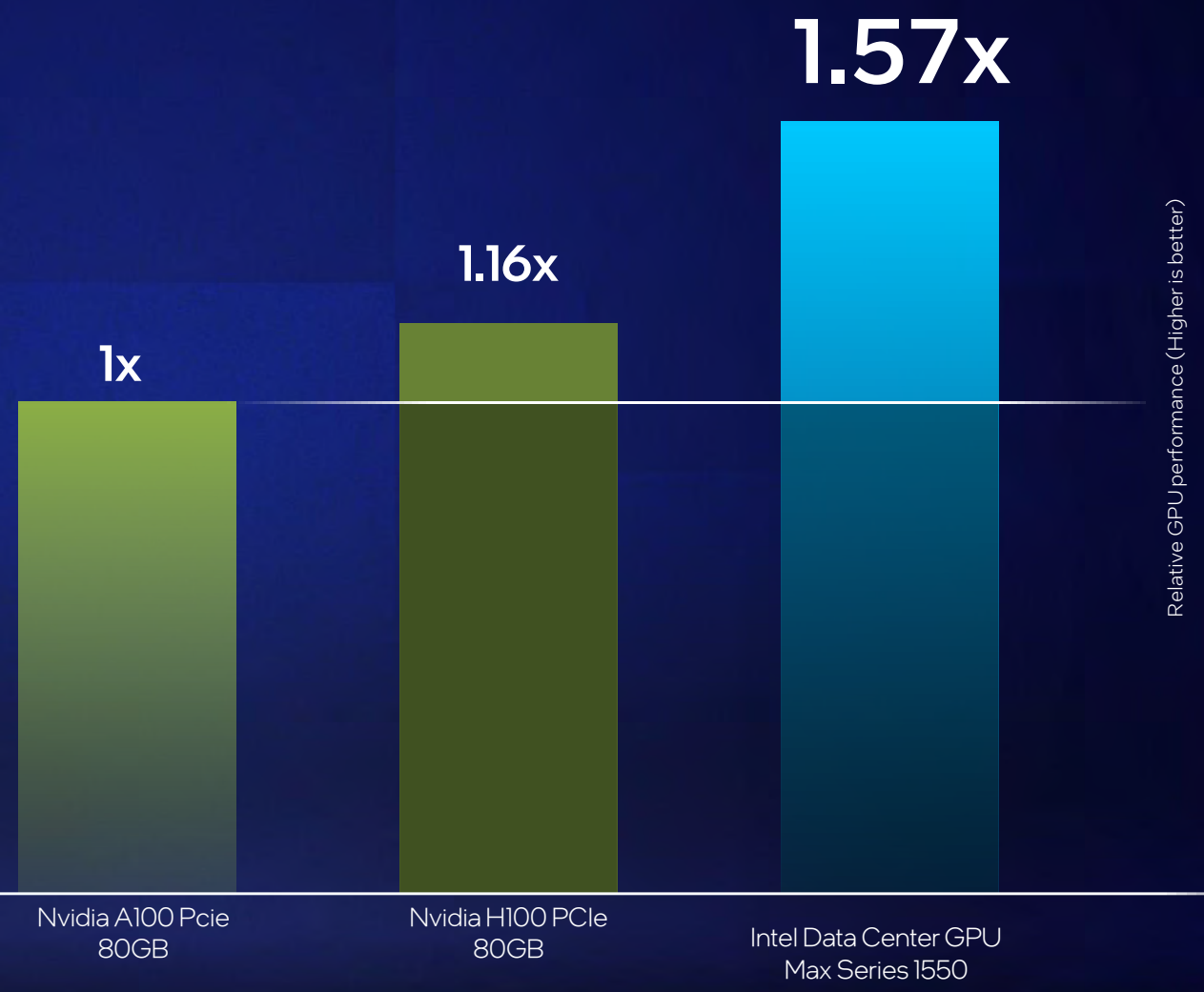
NWChemEx
90 Node performance



Accelerating Fusion Plasma Modelling

WDMApp GENE performance

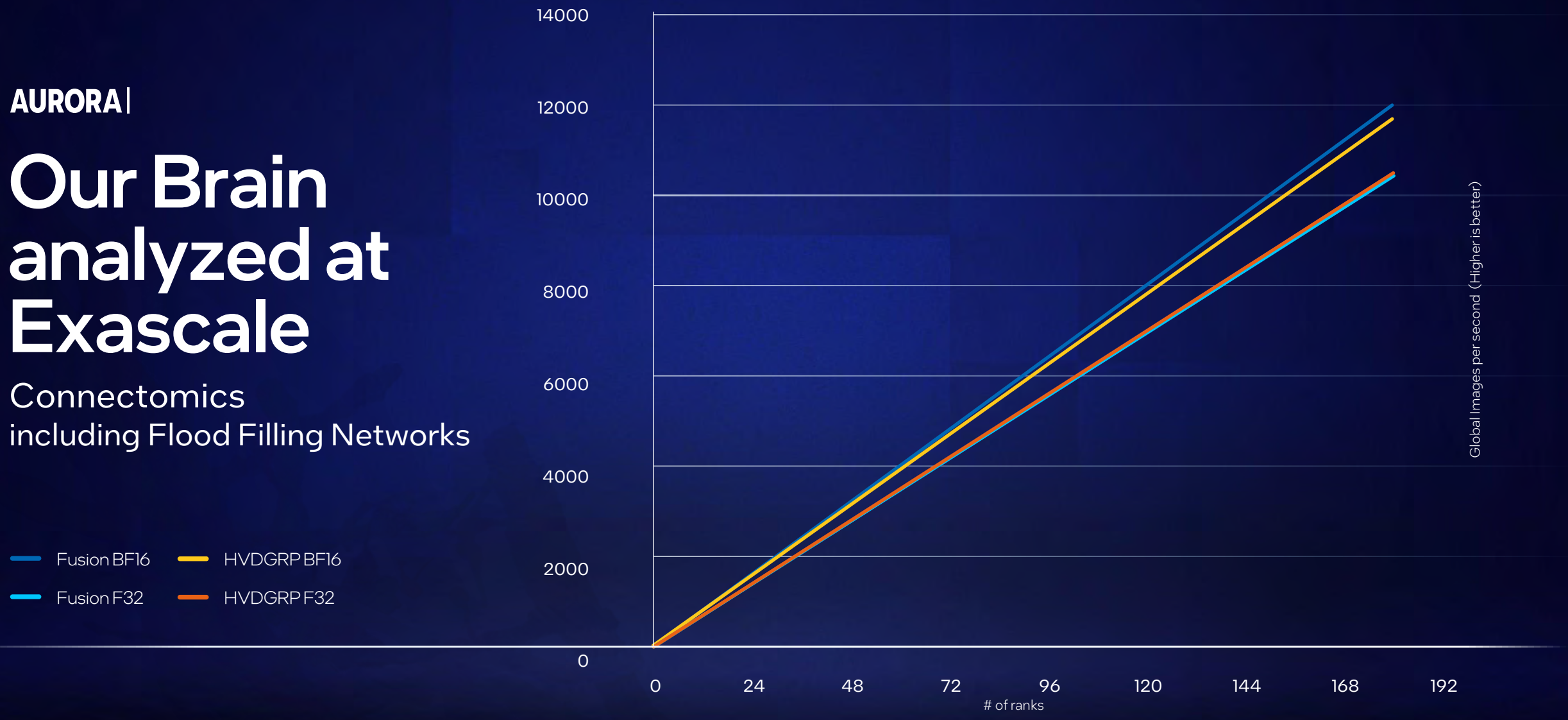
(Single – GPU)



AURORA

Our Brain analyzed at Exascale

Connectomics including Flood Filling Networks



intel. XEON MAX SERIES

First & only x86 CPU with HBM

Choose the right memory configuration for your needs

64GB
HBM2e
4 stacks of 16GB

Up to 220GF/s
HPCG

Up to 2GB
HBM per Core

Memory Modes

HBM Only
Bootable from HBM
No code change

HBM | ~~DDR~~

HBM Flat
2 Memory Regions
SW Optimization Needed

HBM | DDR

HBM Caching
HBM as cache for DDR
No code change

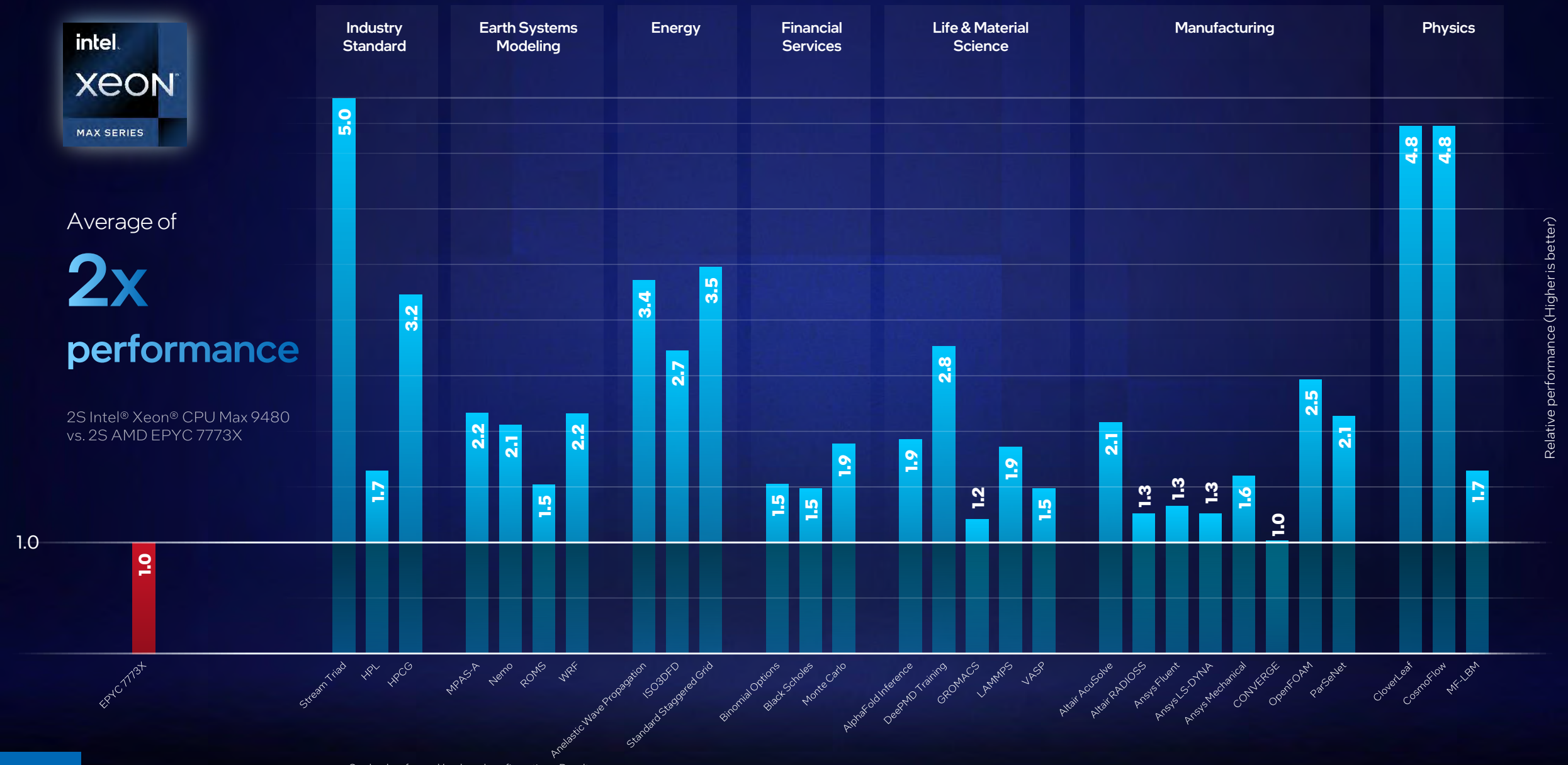
HBM — DDR

See backup for workloads and configurations. Results may vary.

Average of

2x performance

2S Intel® Xeon® CPU Max 9480
vs. 2S AMD EPYC 7773X



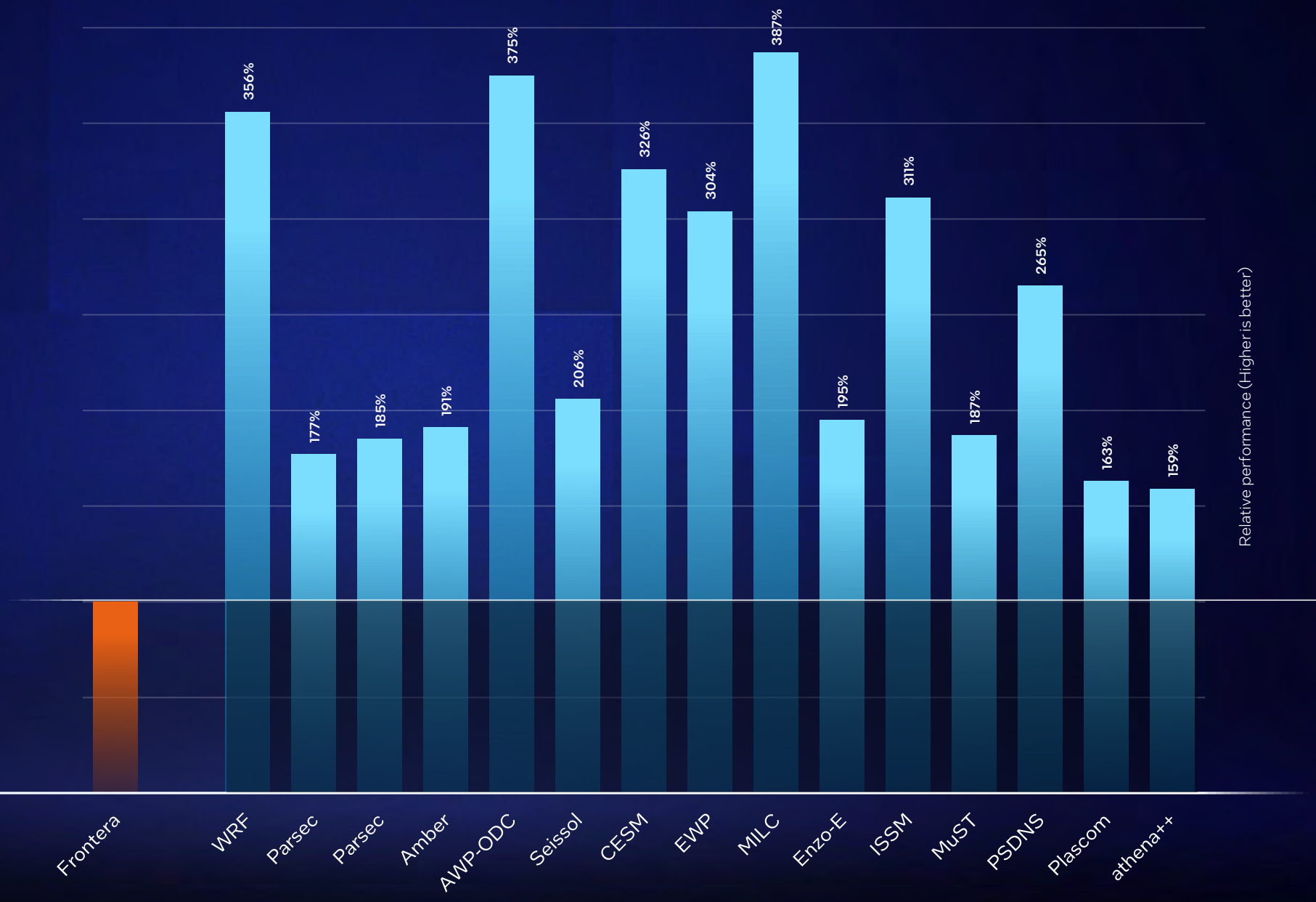
See backup for workloads and configurations. Results may vary.
This offering is not approved or endorsed by OpenCFD Limited, producer and distributor of the OpenFOAM software via www.openfoam.com, and owner of the OPENFOAM® and OpenCFD® trademark
MLPerf™ HPC-AI v0.7 Training benchmark Performance. Result not verified by MLCommons Association. Unverified results have not been through an MLPerf™ review and may use measurement methodologies and/or workload implementations that are inconsistent with the MLPerf™ specification for verified results. The MLPerf™ name and logo are trademarks of MLCommons Association in the United States and other countries. All rights reserved. Unauthorized use strictly prohibited. See www.mlcommons.org for more information

HPC code 252% faster

on average* with HBM

TACC
TEXAS ADVANCED COMPUTING CENTER

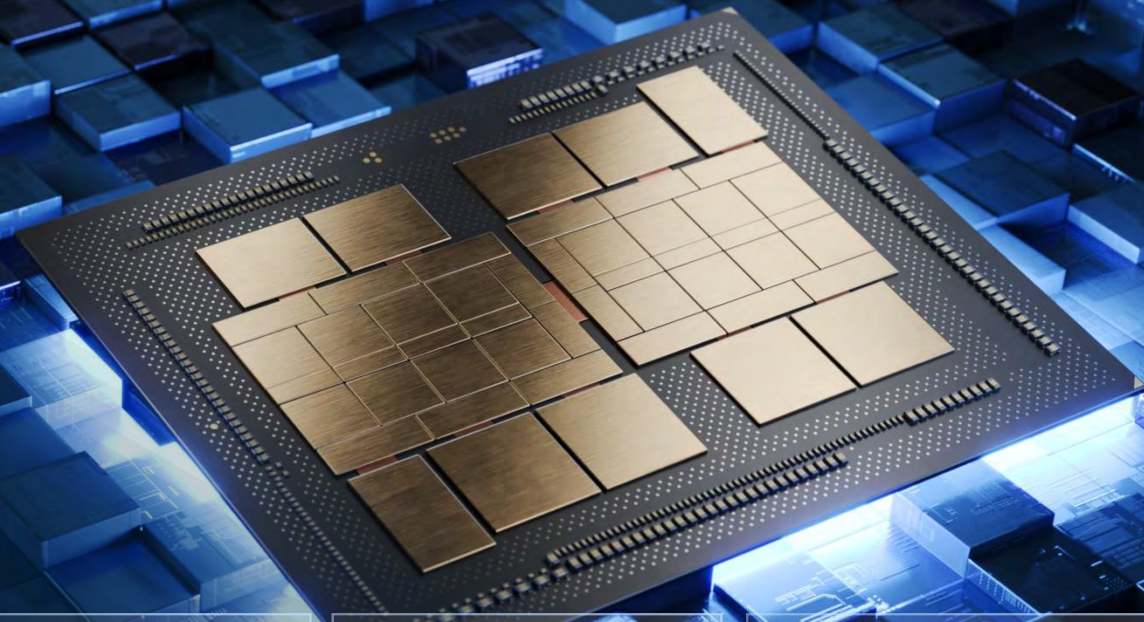
On Xeon Max Series CPU



Visit www.intel.com/performanceindex for workloads and configurations. Results may vary

intel.
DATA CENTER
GPU
MAX SERIES

Intel® Data Center GPU Max Series



Up to
128
Xe HPC Cores

52TF
Peak FP64 Throughput

839TF
Peak BF16 Throughput

Up to
128 GB
HBM2e Memory

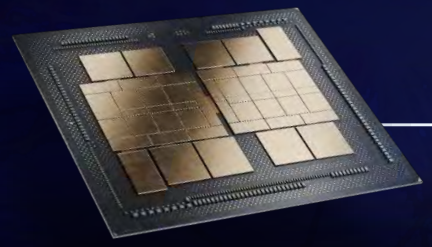
16
Xe Links

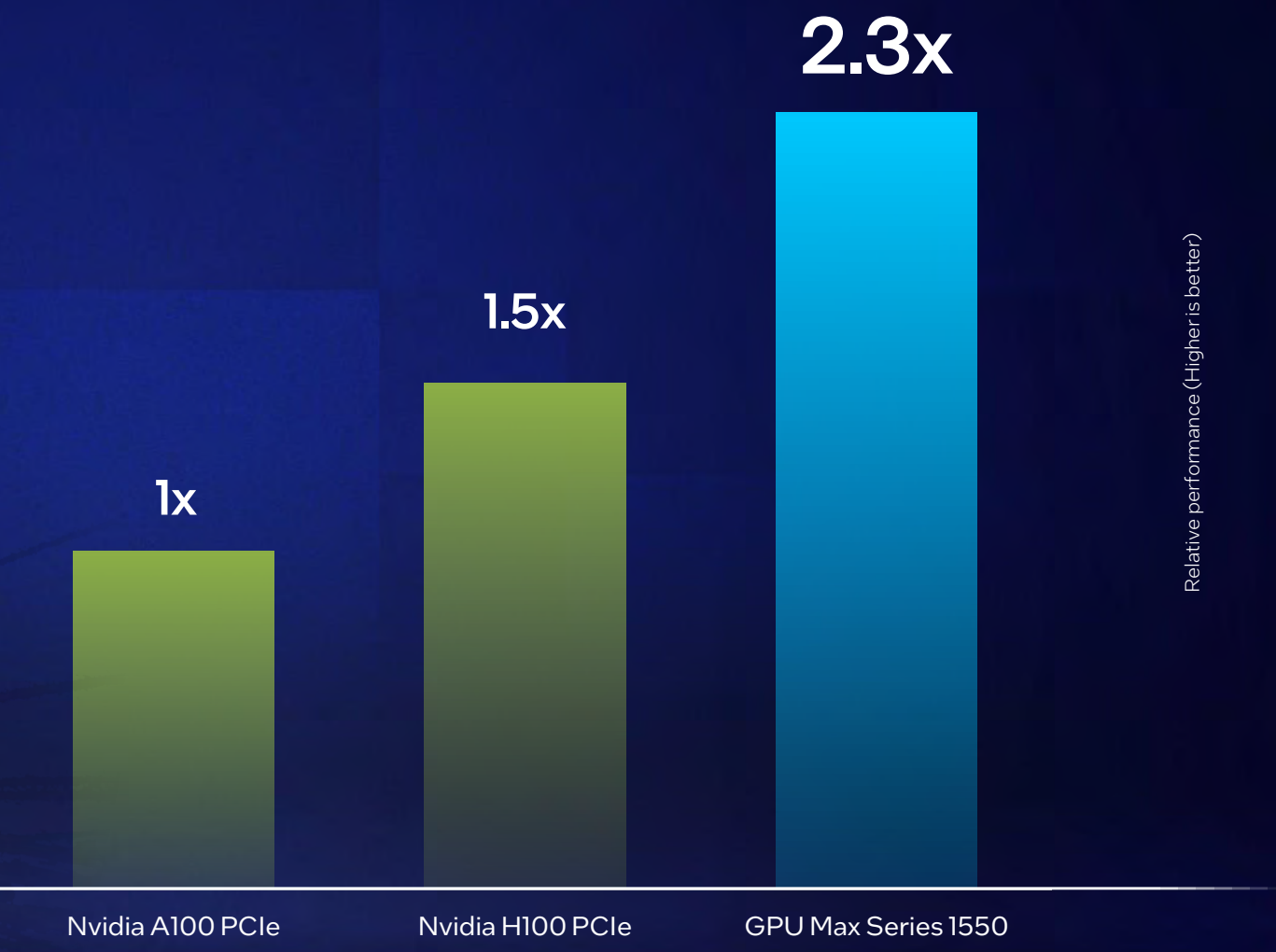
976 GB/s
GPU-to-GPU comms via Xe Links

intel.

Uncovering "Particle Paths" Faster

3D-GAN (BS= 256)

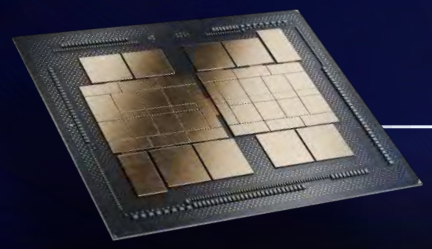


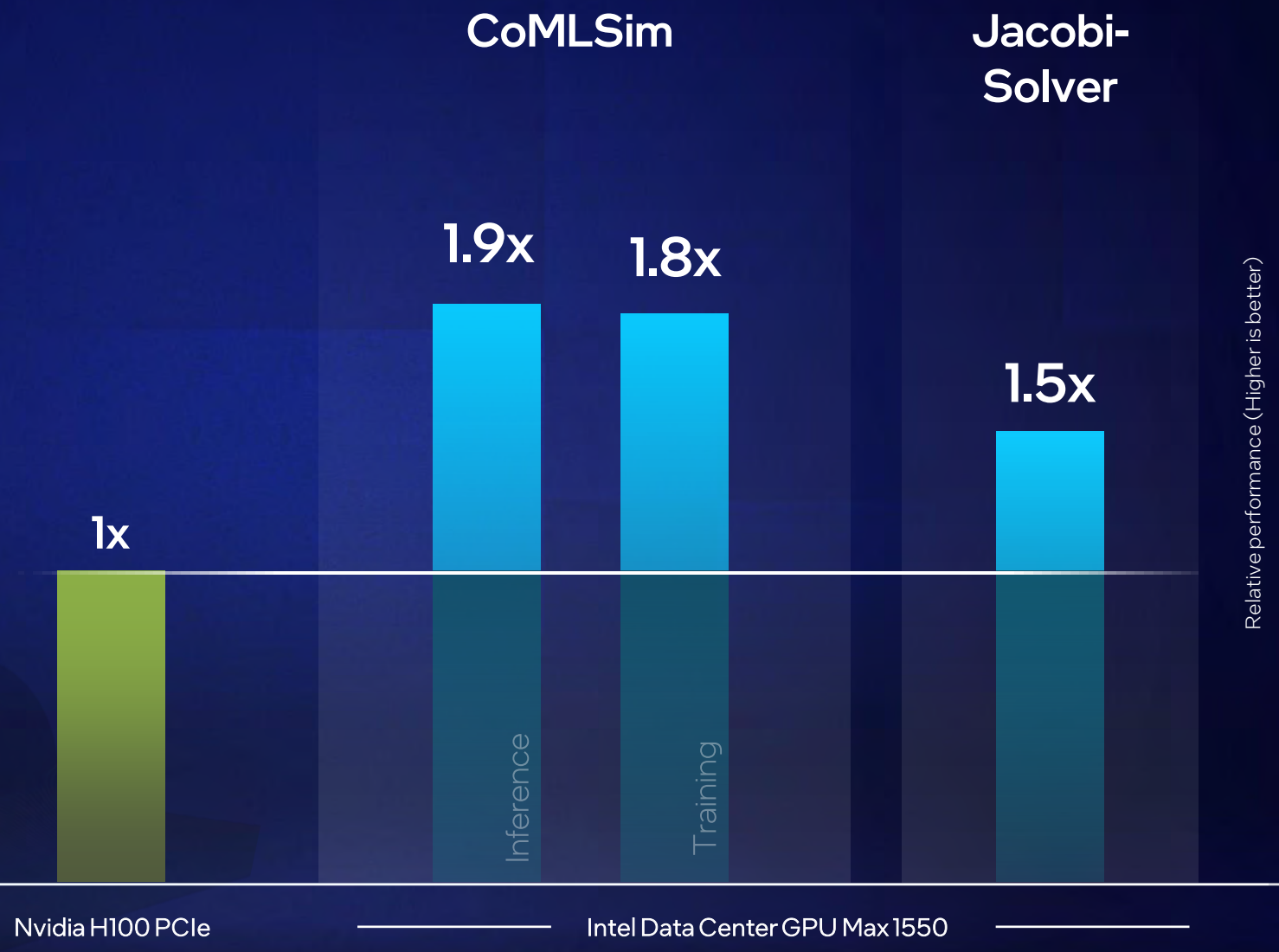


Visit www.intel.com/performanceindex for workloads and configurations. Results may vary

AI powers simulation at top speed

Ansys

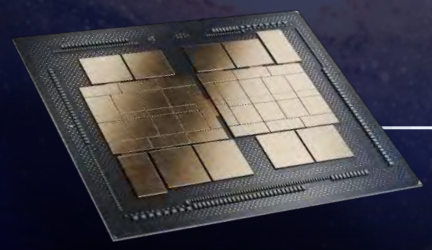


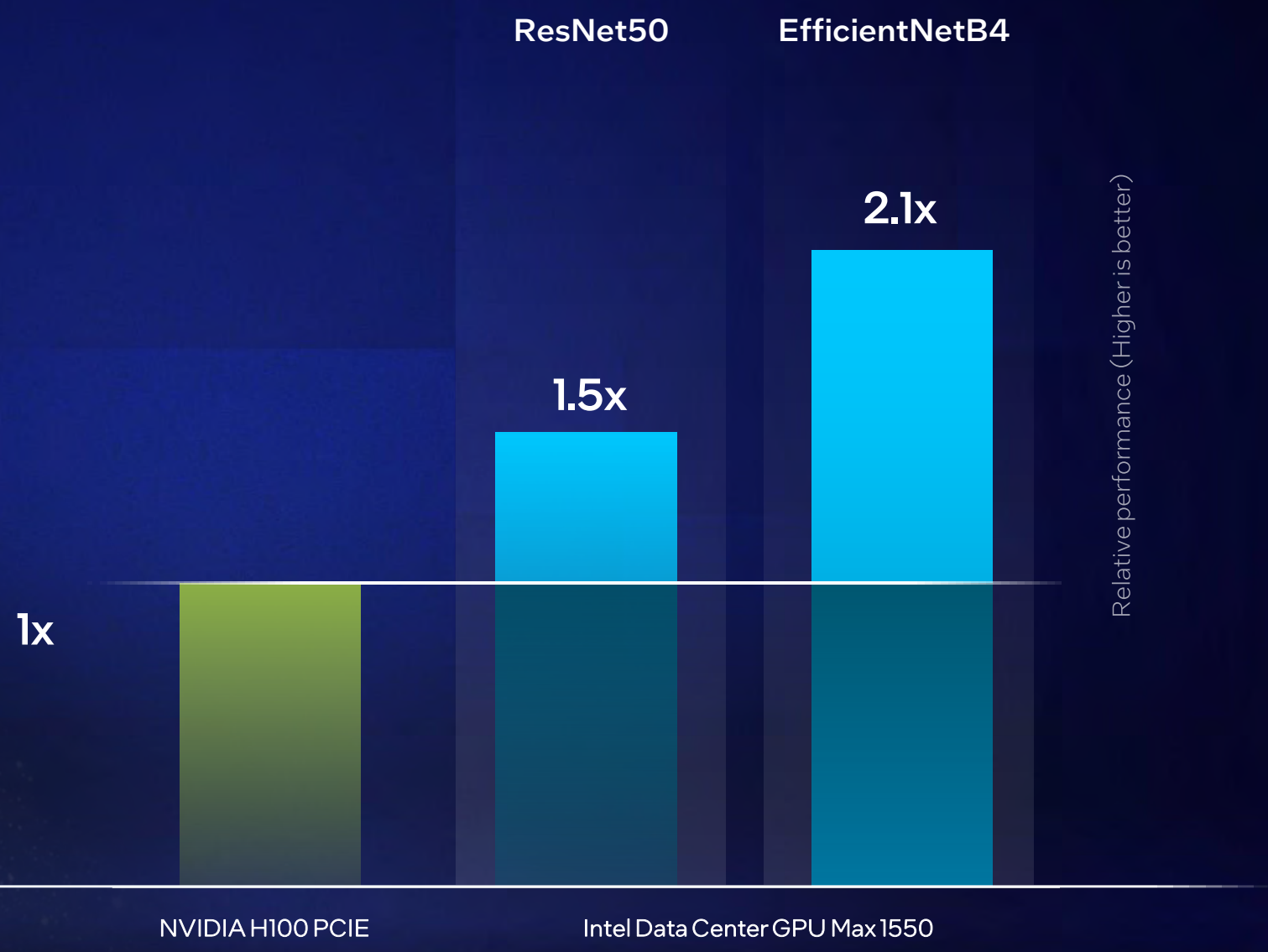


Visit www.intel.com/performanceindex for workloads and configurations. Results may vary

Classifying Galaxy Types at "Lightspeed"

on DeepGalaxy



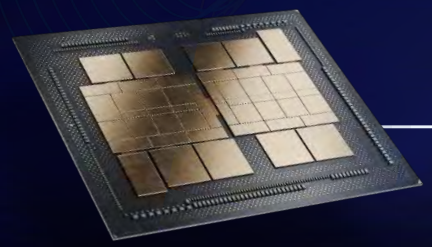


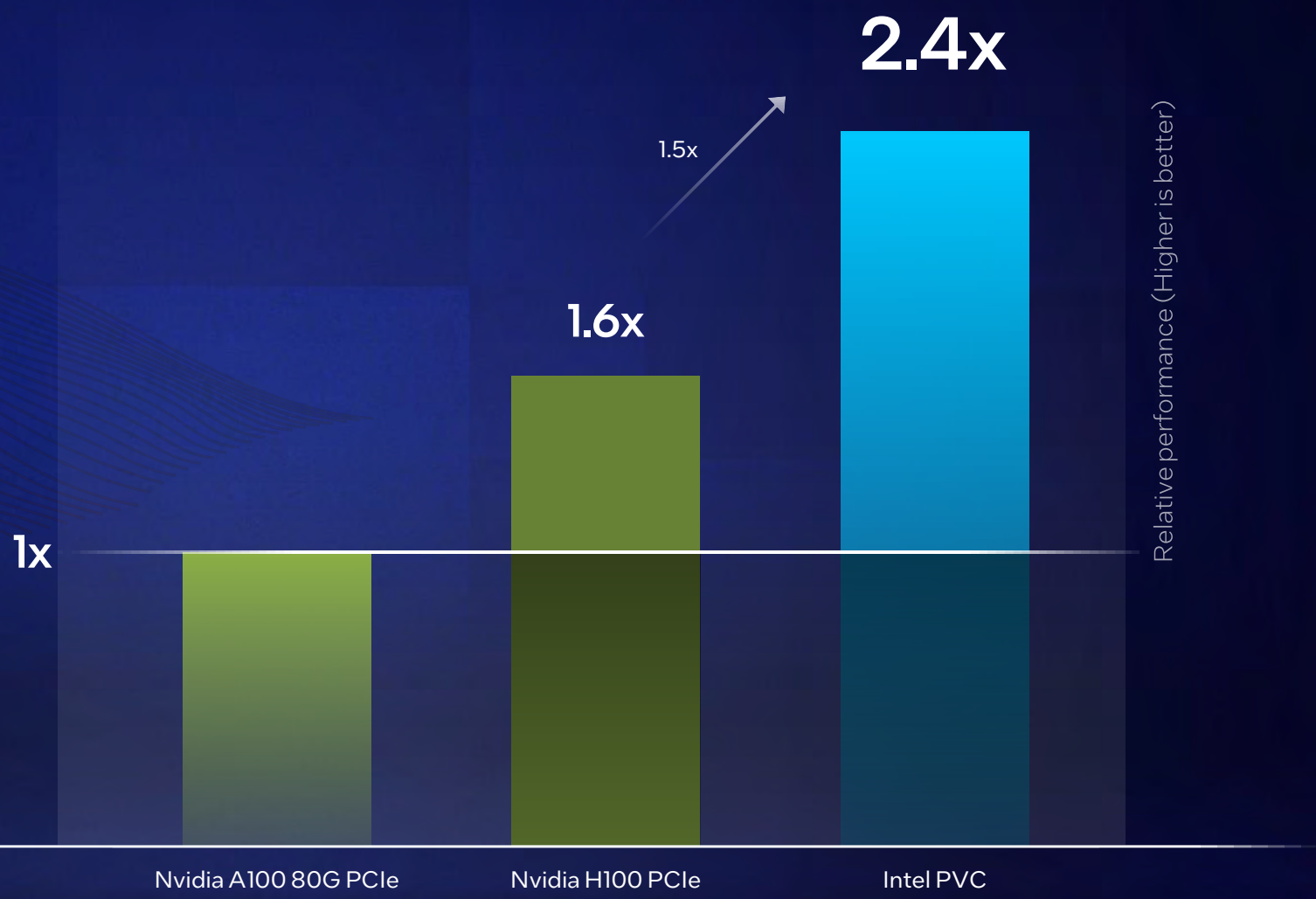
See backup for workloads and configurations. Results may vary.

intel. DATA CENTER GPU MAX SERIES

Train Credit Option Pricing Models. Faster.

Riskfuel

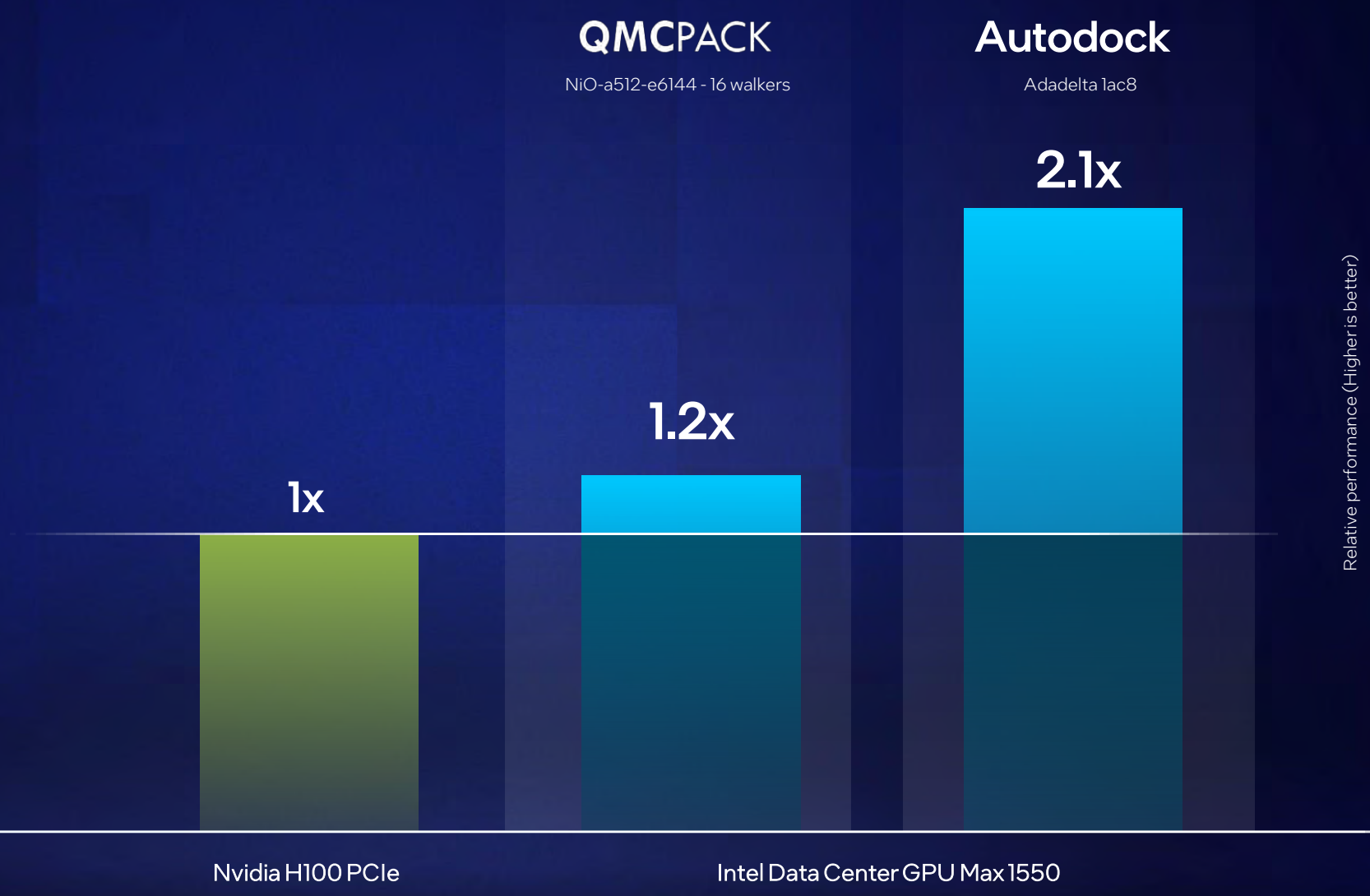


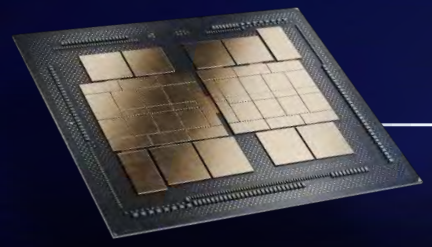


Visit the SuperComputing 22 page at intel.com/performanceindex for workloads and configurations. Results may vary

intel.

Science simulated faster than ever



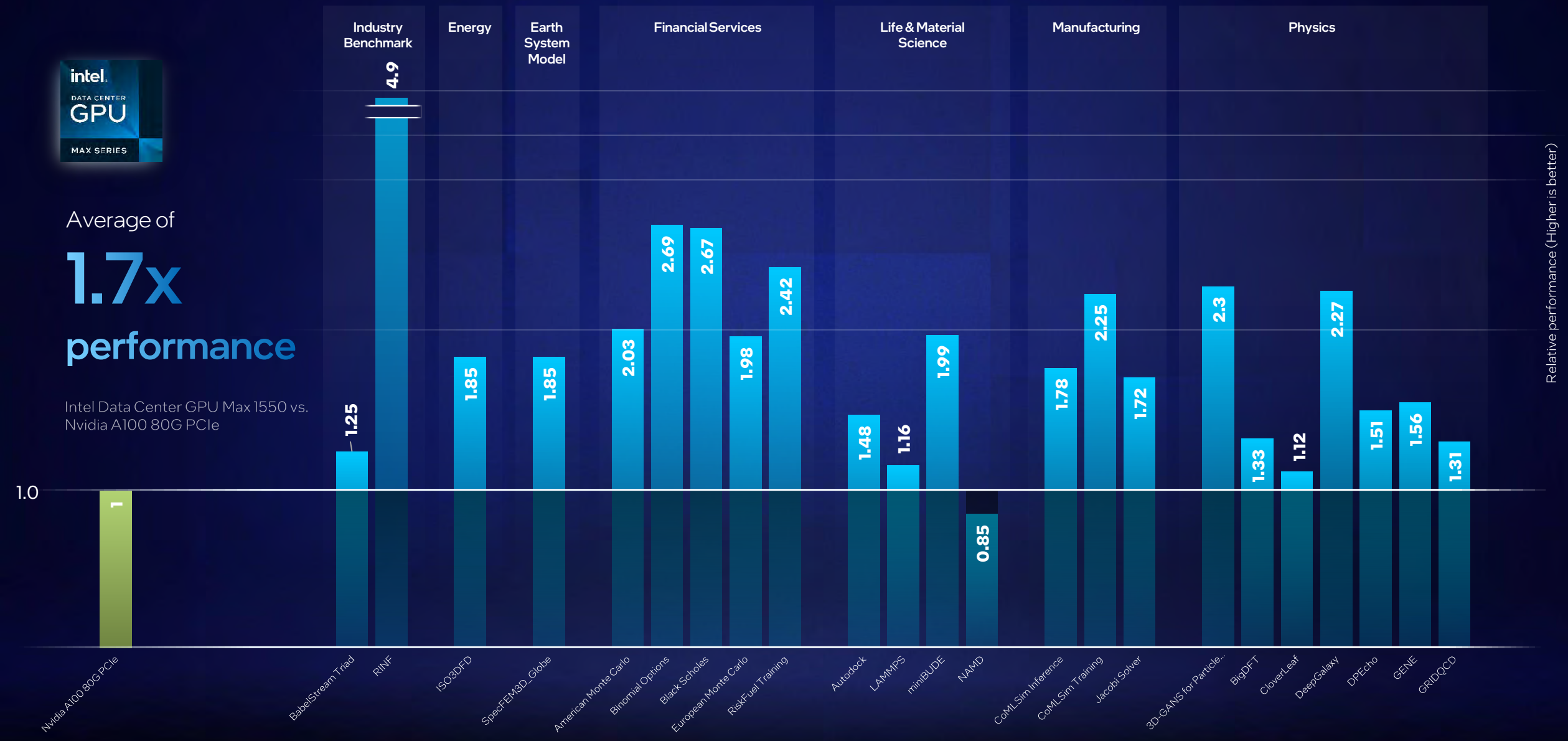


Visit www.intel.com/performanceindex for workloads and configurations. Results may vary

Average of

1.7x performance

Intel Data Center GPU Max 1550 vs. Nvidia A100 80G PCIe



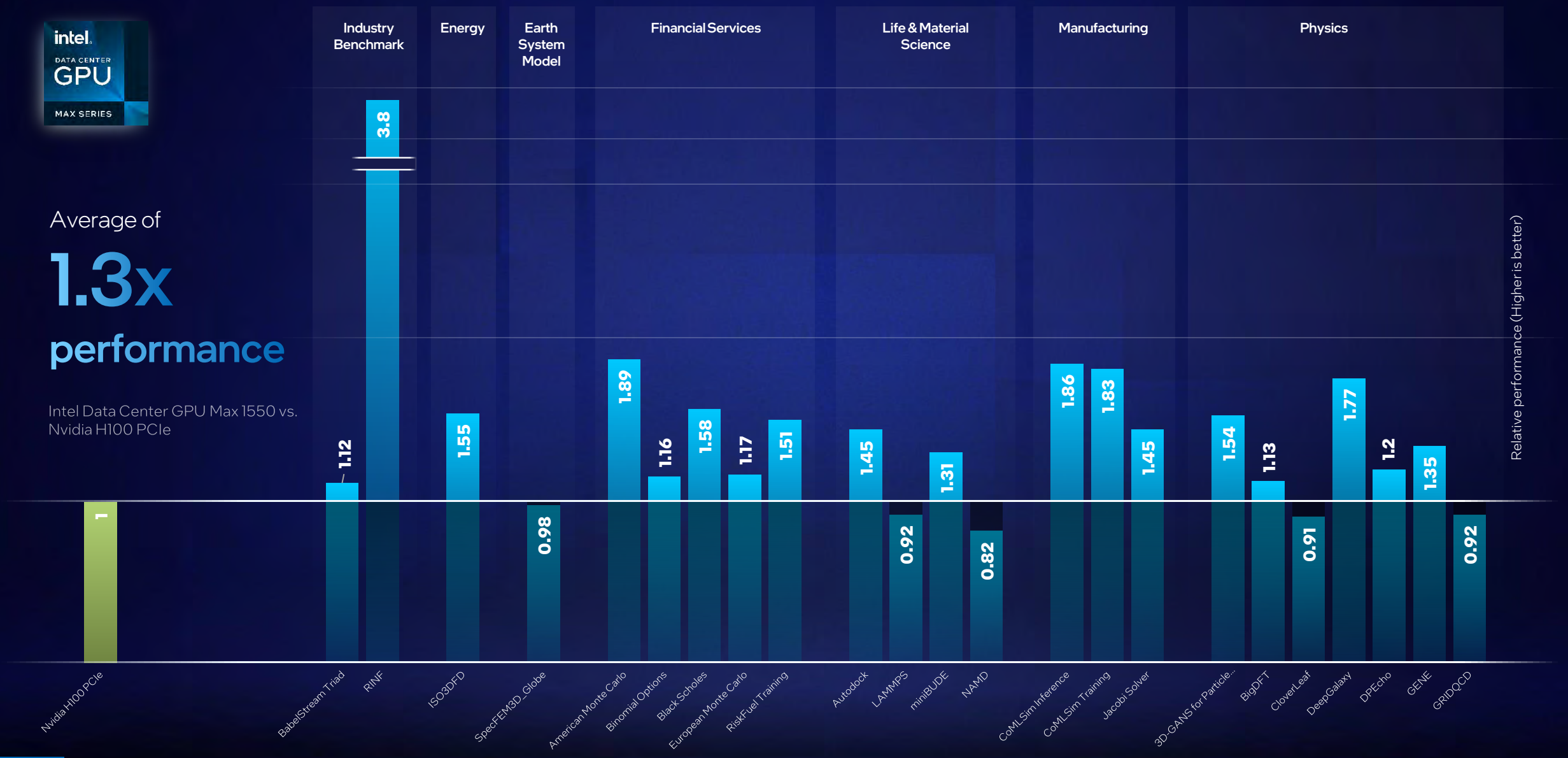
See backup for workloads and configurations. Results may vary.

intel. DATA CENTER GPU MAX SERIES

Average of

1.3x performance

Intel Data Center GPU Max 1550 vs. Nvidia H100 PCIe



intel.

Visit www.intel.com/performanceindex for workloads and configurations. Results may vary

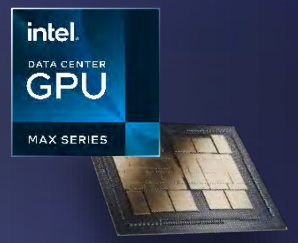
A lot has happened in the last year…

Leading performance on AI & HPC apps

Strengthened the Unified Software layer

Scale | Open | Trusted | Choice

Delivered Leadership Compute for all HPC & AI Needs



*Conceptual Representation

intel.

intel®

oneAPI *and* C++ with SYCL:

Innovation through Abstraction for ALL hardware – CPUs and accelerators

James Reinders

intel®

Together – we are at the forefront of helping

support programming to target
ALL VENDORS and ALL ARCHITECTURES

SOFTWARE CHALLENGE

A New Golden Age for Computer Architecture

High-level, domain-specific languages and architectures, freeing architects from the chains of proprietary instruction sets, along with demand from the public for improved security, will usher in a new golden age for computer architecture.

Diverse and evolving workloads enable hardware innovation

Source: John L. Hennessy, David A. Patterson, Communications of the ACM
https://cacm.acm.org/magazines/2019/2/234352-a-new-golden-age-for-computer-architecture/fulltext

- seek to help software be open to *all* CPUs and *all* accelerators

Aurora innovation – push boundaries of OPEN for everyone's benefits

Multivendor & Multiarchitecture

Builds on learnings from **Kokkos** – high HPC value – C++ focus

C++ with SYCL grew from OpenCL learnings – C++ focus

oneAPI grows from learnings of Kokkos, SYCL, OpenCL, CUDA – foundational – C++ focus

OpenMP and **MPI** still alive, well, and very important – just not today's talk topic

- seek to help software be open to *all* CPUs and *all* accelerators

Why Intel?

- Intel has great products to offer: CPUs, GPUs, FPGAs, and more

- Intel has great manufacturing to offer for ALL.

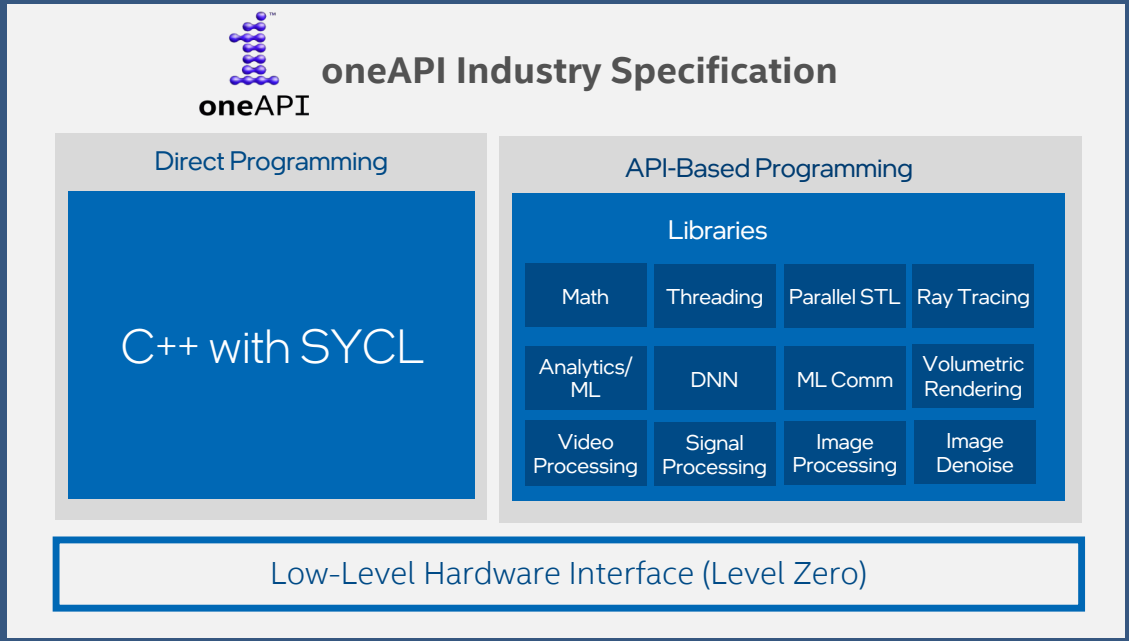
oneAPI: One Name, Two Distinct Objectives

oneAPI

- Newly Independent Organization
- Founding members will announce more later this year
- Open industry specification
- Open-source repo and development
- Community driven
- Supports multi-vendor implementation

oneAPI

- Product support from Intel and Codeplay
- Intel's implementation
- Toolkits optimized for Intel HW
- Available for free download



- Open, standard language: C++ with SYCL
- clang + LLVM – contributions and benefits
- Standardized interfaces for common libraries
- Standardized hardware interface

*SYCL and the SYCL logo are trademarks of the Khronos Group Inc. in the U.S. and/or other countries

Why C++ with SYCL?

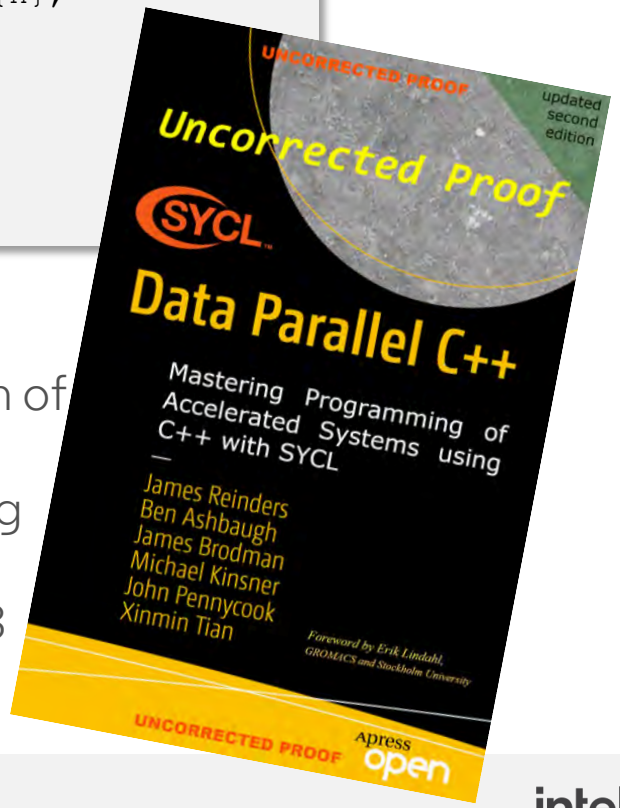
```
sycl::queue q(cpu_selector{});

auto A = sycl::malloc_shared<float>(n, q);
auto B = sycl::malloc_shared<float>(n, q);

q.parallel_for( sycl::range<1>{n},
  [=] (sycl::id<1> i) {
      B[i] += A[i] * A[i];
  }
).wait();
```

```
sycl::queue q(gpu_selector{});

auto A = sycl::malloc_shared<float>(n, q);
auto B = sycl::malloc_shared<float>(n, q);

q.parallel_for( sycl::range<1>{n},
  [=] (sycl::id<1> i) {
      B[i] += A[i] * A[i];
  }
).wait();
```

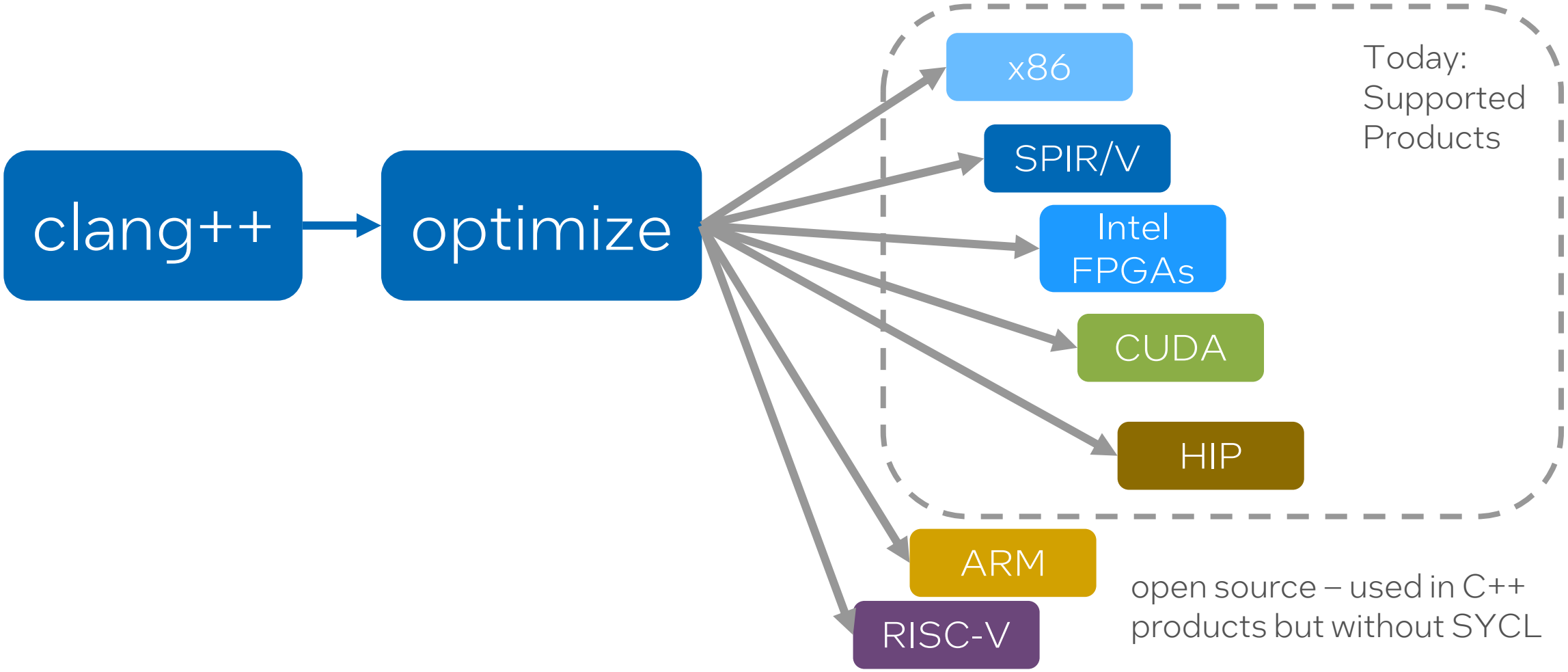
*Example uses static **targets in bold**, but can be programmed to be dynamic as well

- Standard C++17 aids time to developer productivity
- Syntax for accelerators (device selection, offload, memory transfer)
- Unified shared memory
- Single source (host and device code)
- Multi-Architecture (CPU, GPU, FPGA, and other targets)
- Stack based on standards and open specifications (CLANG, LLVM, SPIR-V, Level Zero)

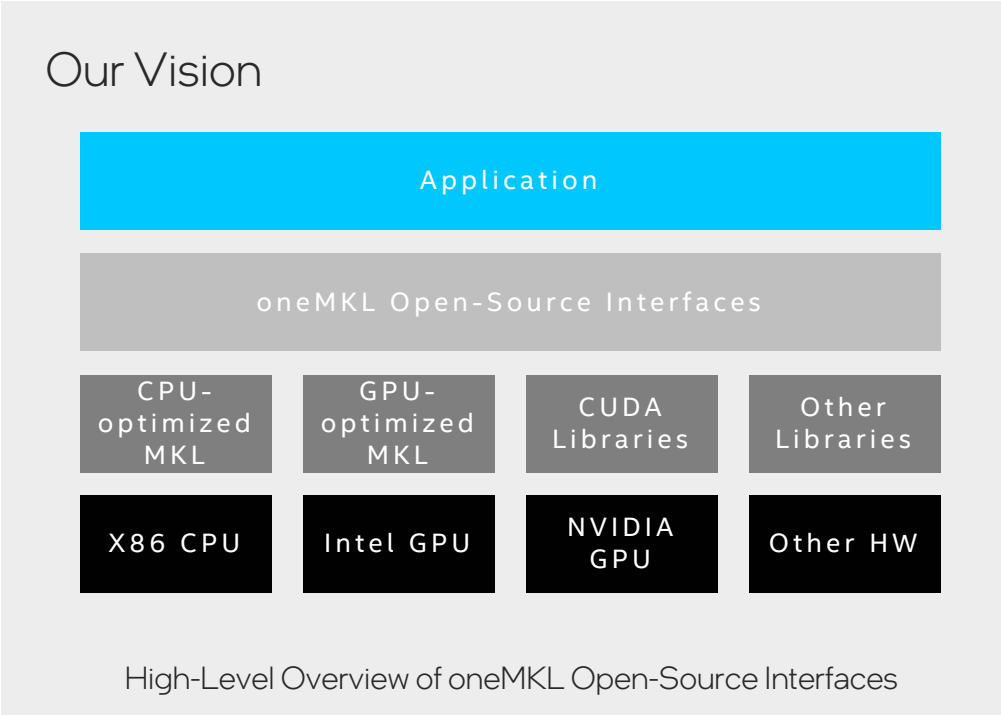
2nd Edition of Book coming by Q4 '23



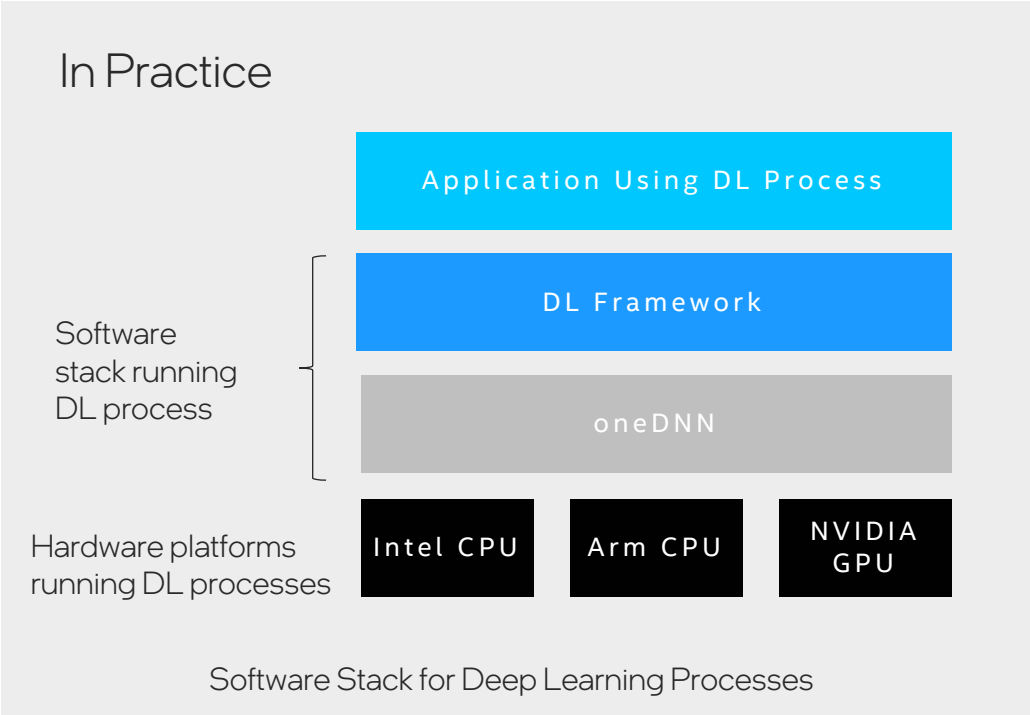
open source is wonderful
C++ with SYCL in LLVM – uses native backends



Standardized Library Interfaces



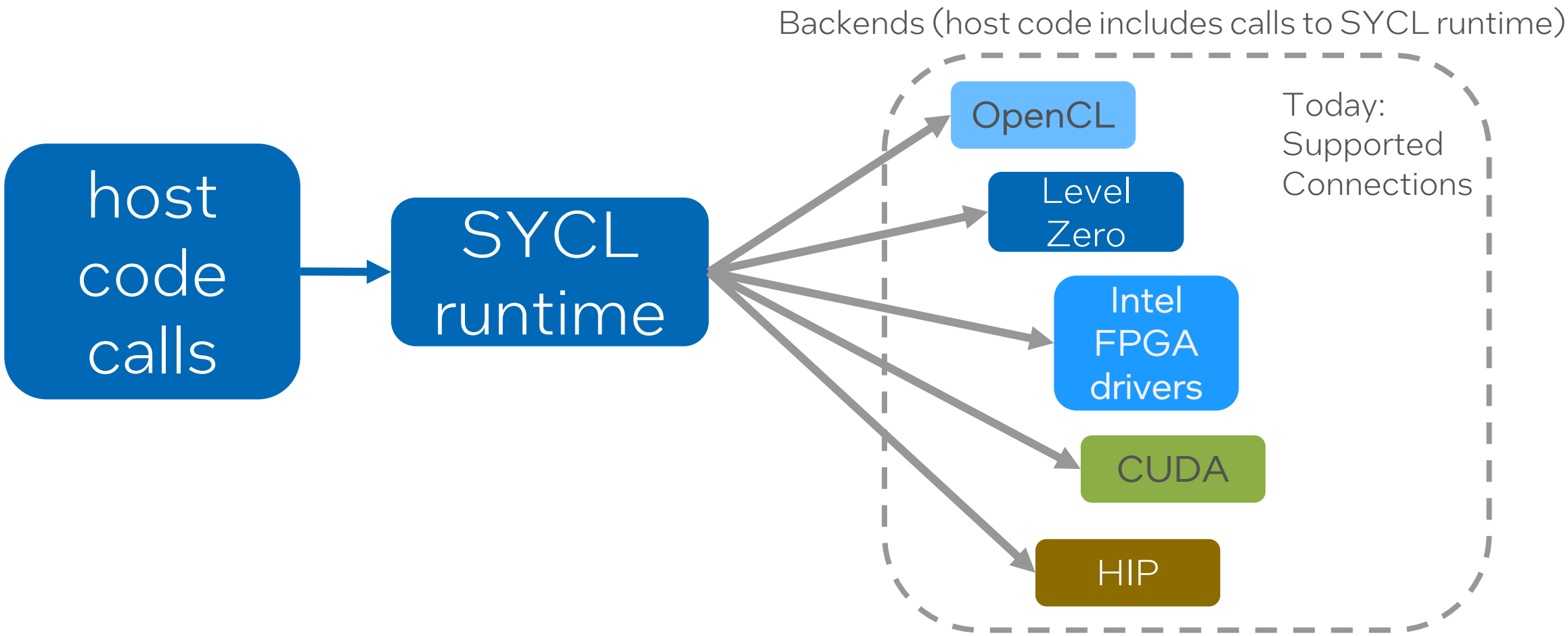
Source: https://www.intel.com/content/www/us/en/developer/articles/technical/a-vendor-neutral-path-to-math-acceleration.html#gs.v4bfu6



Source: https://blog.fltech.dev/entry/2020/11/19/fugaku-onednn-deep-dive-en

- Faster time to applications
- Ability for HW companies to provide unique value-add underneath standard interfaces

C++ with SYCL in LLVM runtime flexible connections



Status and How to Join In the Fun

vision: true multivendor and multiarchitecture

- github – open source
  - oneAPI is a collection of activities generally grounded in open-source projects (with additions planned)
  - LLVM, libraries, tools, and the specification itself
  - input and contributions are always welcome
- Adoption
  - oneAPI brings advantages that are leading many to evaluate and commit to oneAPI adoption
- Intel oneAPI support – pioneering multivendor "plugin" in product offerings
  - In late 2022, Intel introduced a "plug in" model for its C++ (with SYCL) compiler that allows seamless (no overhead) addition of non-Intel support into the same compiler, with merged single-binary outputs supporting multiple vendors
  - Codeplay announced product support for NVIDIA and AMD GPUs
- oneAPI specification – governance – https://oneapi.io
  - oneAPI is transitioning to a fully independent process
    - Intel shepherded oneAPI in early years, with independent technical advisory boards (all meeting notes are openly online)
    - In late 2022, Intel announced formation of independent steering committees
    - Organization interested in being "founding members" are participating in decisions on the exact details of this independent foundation
      - Goal: formalize governance and membership, agreement, and announce final decisions before end of 2023

oneAPI Ecosystem Support

These organizations support the oneAPI initiative for a single, unified programming model for cross-architecture development.
It does not indicate any agreement to purchase or use of Intel's products. *Other names and brands may be claimed as the property of others.

Summary

- oneAPI: multi-vendor, multi-architecture programming for accelerators
- Standards and open specification based
  - SYCL language
  - Standard library interface
  - LLVM/CLANG, SPIR-V, Level Zero
- Available today in both open-source GitHub and as finished Intel product

Visit oneapi.io for specifications, source repository

Please join the community, provide input

Intel platforms – download for free at software.intel.com

intel®

ALCF Webinar Developer's Session

Software: oneAPI Toolkits and Programming Models for Aurora

Xinmin Tian, Intel Fellow
Intel Corporation, June 21, 2023

intel®

Notices & Disclaimers

Intel technologies may require enabled hardware, software or service activation. Learn more at intel.com or from the OEM or retailer.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804. https://software.intel.com/en-us/articles/optimization-notice

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. See backup for configuration details. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Agenda

- oneAPI Software Stack: The Big Picture
- C/C++/Fortran OpenMP Programming Model and Compilers
  - New features
  - Performance on SPR
  - Performance on PVC
- SYCL Programming Model and Compilers
  - New features
  - SYCL 2020 conformance
  - Performance on PVC
  - SYCL Cross HW-IP Performance
- Summary
- Call for Actions

oneAPI: The Big Picture

SW Contract: Open, Free, Everywhere, Forever	Applications, Services & Solutions		
	Middleware, Frameworks & Runtimes		
LEVEL ONE Abstraction	Direct Programming Languages SYCL/OpenMP Languages | Ecosystem Languages	API Programming Libraries oneAPI Libraries | Ecosystem & Intel Libraries	**Tools** Port, Compile, Debug, Analyze Ecosystem & Intel Libraries | Ecosystem Tools
LEVEL ZERO Abstraction	Hardware Abstraction Layer oneAPI Level Zero | Ecosystem HALs		
Freedom of **Choice** in Hardware	Architecture-Specific Instruction Set x86 ISA | X^e GPU ISA | FPGA ISA | ASIC ISA | ARM ISA | NV GPU ISA (PTX)		

Specification and more information: https://spec.oneapi.com

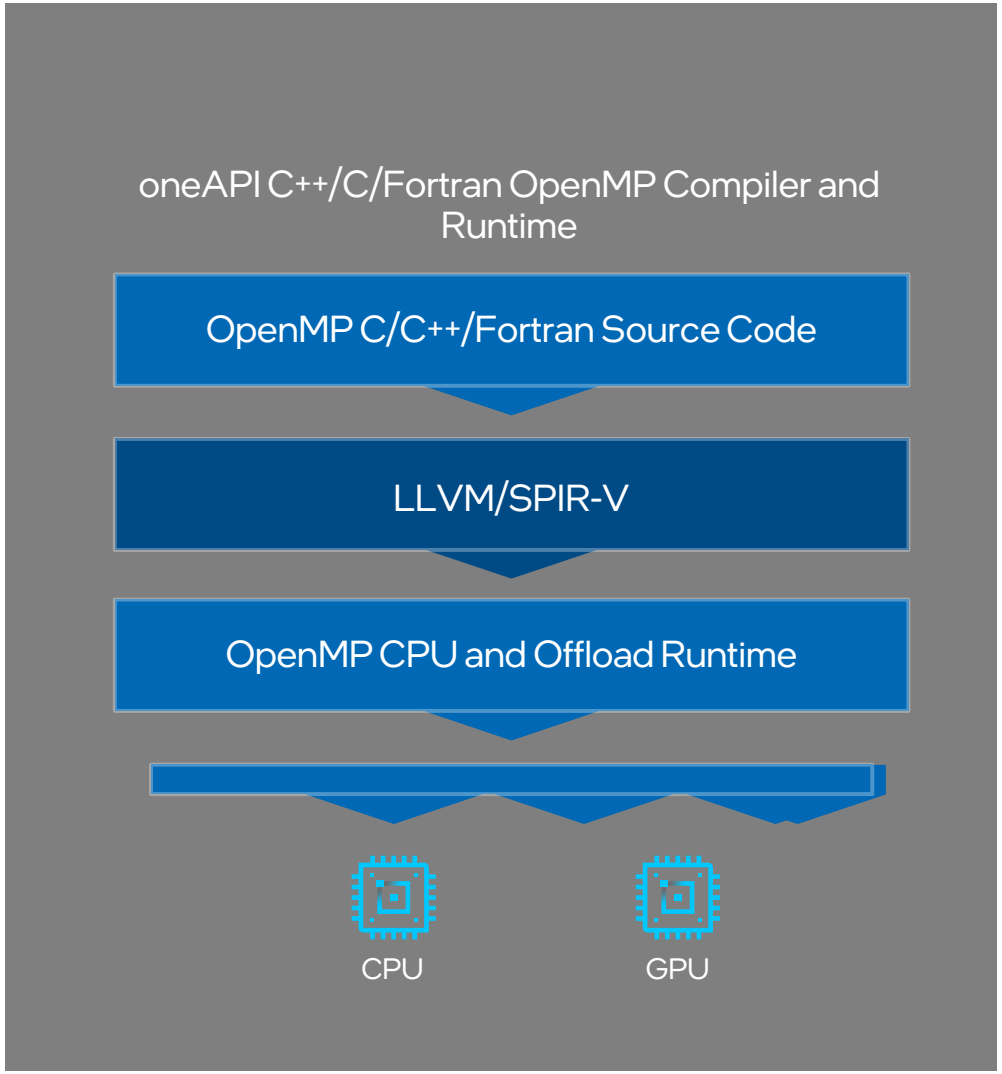
Intel® C++/SYCL & Fortran OpenMP* Compilers

Parallel Programming Productivity & Performance

Compiler to deliver uncompromised parallel programming productivity and performance across CPUs and GPU accelerators

- Allows code reuse across hardware targets, while permitting custom tuning for a specific GPU accelerator
- Delivers C/C++/SYCL and Fortran OpenMP productivity benefits, by supporting C++20 common and familiar C, C++ constructs, Fortran 2008 and 2018 standards
- Support features in OpenMP 5.1/5.2/TR11 and SYCL2020 data parallelism and heterogeneous programming

Builds upon Intel's decades of experience in architecture and high-performance compilers



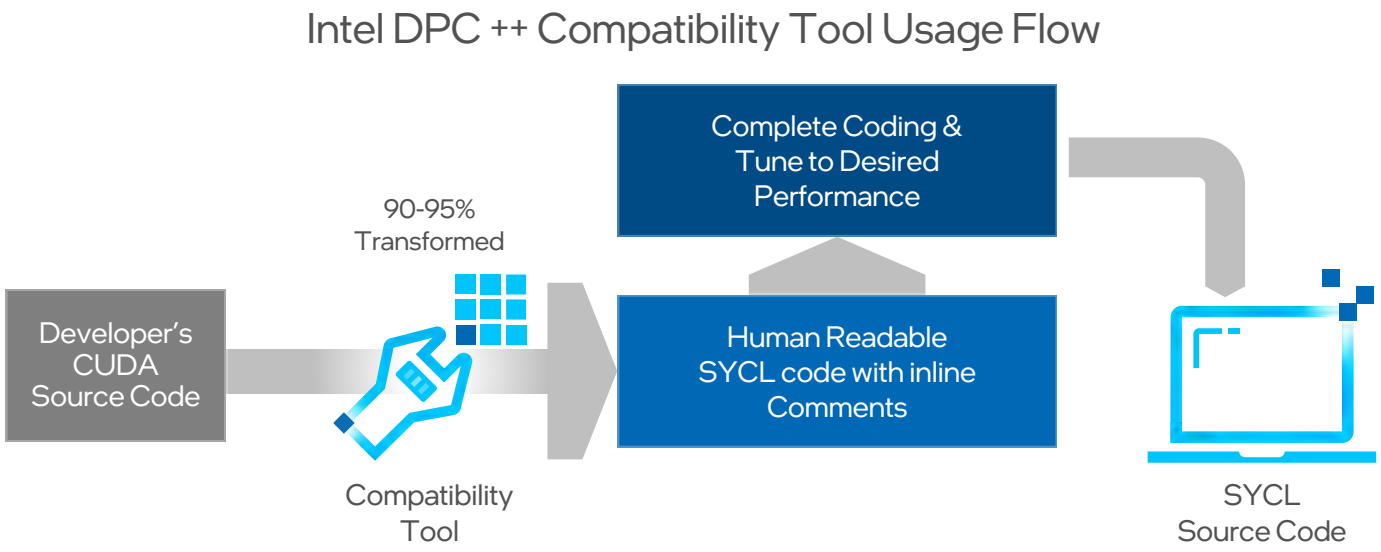
Intel® DPC++ Compatibility Tool

Minimizes Code Migration Time

Assists developers migrating code written in CUDA to SYCL once, generating **human readable** code wherever possible

~90-95% of code typically migrates automatically[1]

Inline comments are provided to help developers finish porting the application



[1]Intel estimates as of September 2021. Based on measurements on a set of 70 HPC benchmarks and samples, with examples like Rodinia, SHOC, PENNANT. Results may vary.

New Features in oneAPI Compiler Releases (2023.x)

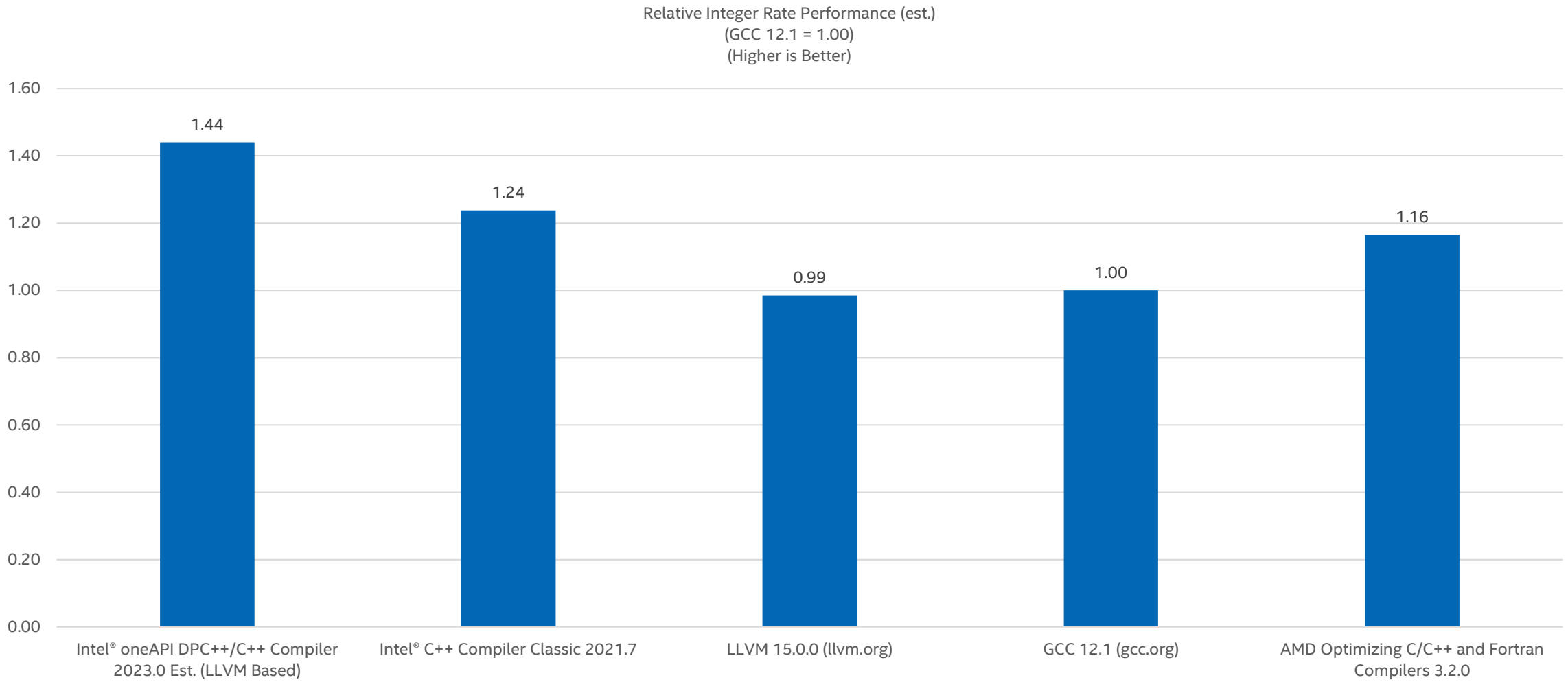
- Just-In-Time (JIT) and Ahead –Of-Time (AOT) Compilations for Intel new Xeon CPUs and Xe GPUs hardware enabling
- OpenMP 5.1/5.2/TR11 features
  - Dispatch (Intel), dispatch, declare variant (subset) declare mapper (C/C++), interop, loop, scope (C/C++), allocate directive/clause, align clause/modifier, nowait for task wait, target in_reduction clause, conditional lastprivate clause, ompx prefetch extension for GPUs, OpenMP SIMD for GPUs, loop tiling.
- C++20, Fortran 2008, Fortran 2018 OpenMP Offloading
- Unified Shared Memory (USM)
- OpenMP and SYCL/DPC++ Composability
- Multi-GPU and Multi-Tile Support
- Asynchronous Offloading
- Optimization Report
- Performance Optimizations (prefetch, tree-like reduction, loop optimizations, HBM, … etc.)

OpenMP C/C++/Fortran Status

- Subset of 5.1/5.2 specification
  - ~93% features of 5.1/5.2 specification
  - ~93% of OpenMP runtime APIs
  - ~95% of OpenMP environment variables

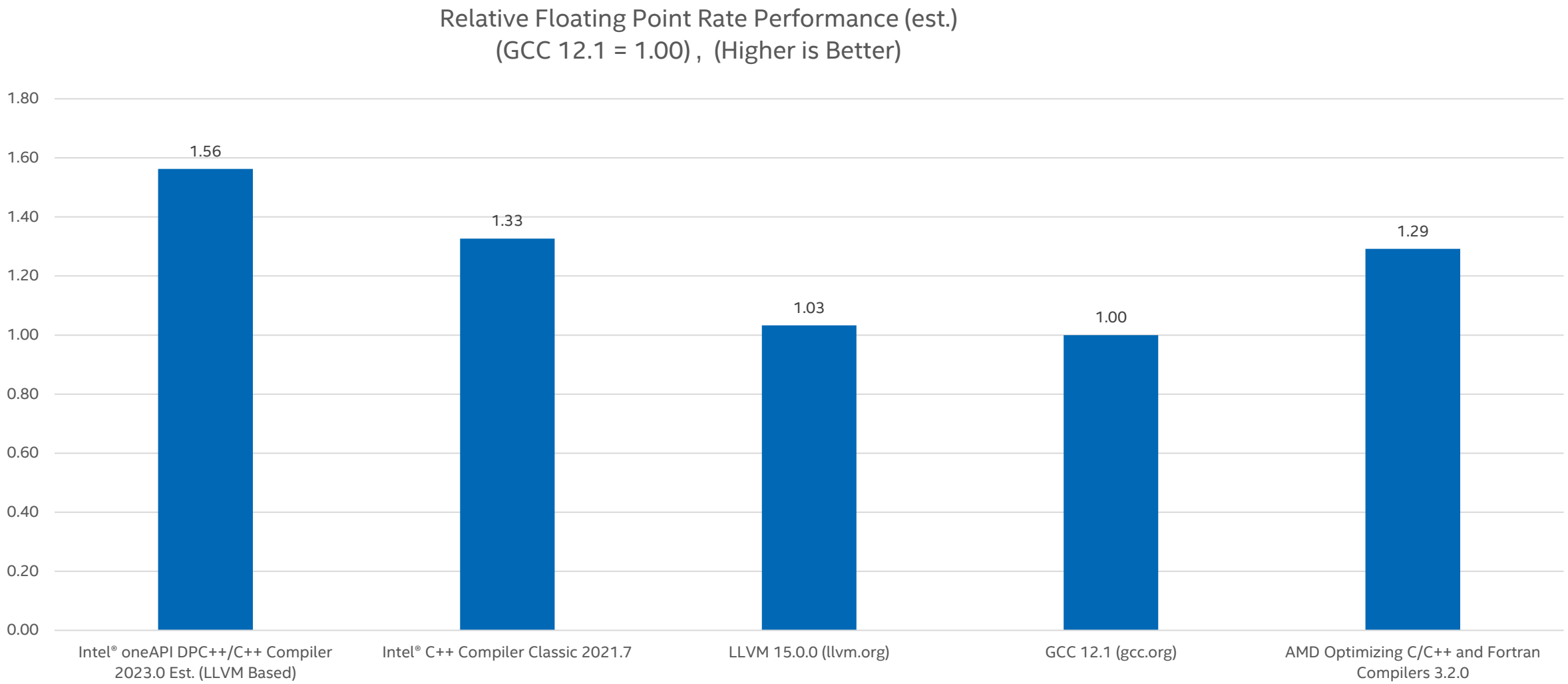
- The focus areas for the next quarter include
  - New application required language features for SPR and PVC performance
  - Compiler quality
  - Application compile-time improvement

Intel® C/C++ Compiler Performance on SPR



Estimated: internal measurement of the geometric mean of the C/C++ workloads from the SPECrate* 2017 Integer suite (base tune)

Intel® C/C++ Compiler Performance on SPR



Estimated: internal measurement of the geometric mean of the C/C++ workloads from the SPECrate* 2017 Floating-point suite (base tune)

PVC Enhanced Compiler Support

- Prefetch extension for PVC
- OpenMP Async-offloading for PVC with helper threads and IMM
- Support for is_device_ptr, use_device_ptr, has_device_addr and use_device_addr
- Ahead-of-Time (AOT) compilation and parallel build
  - icpx -fiopenmp -fopenmp-targets=spir64_gen -fopenmp-device-code-split=per_kernel -Xopenmp-target-backend "-device pvc" test.c -fopenmp-max-parallel-link-jobs=4
- Advisor Tool Refined Compiler Integration
  - Offloading profitability analysis

SYCL Everywhere: (Khronos Highlights)

- SYCL defines abstractions to enable heterogeneous device programming, an important capability in the modern world which has not yet been solved directly in ISO C++.
- A major goal of SYCL is to enable different heterogeneous devices to be used in a single application — for example simultaneous use of CPUs, GPUs, and FPGAs.
- SYCL uses generic programming with C++ templates and generic lambda functions to enable higher-level application software

There's a significant effort readying Aurora for use with oneAPI/SYCL.

Machine	GPU Programming Models	
	CUDA, SYCL	ALCF Polaris
	CUDA, SYCL	NERSC Perlmutter
	HIP, SYCL	OLCF Frontier
	SYCL	ALCF Aurora

*OpenMP has been excluded for context

SYCL 2020 Conformance

- Accessors error checking. Spec: 4.7.6.9. Buffer accessor for commands
- Accessor iterators, zero-dimensional `accessor`
- Constexpr vec constructors
- Identity-less reduction
- `marray` overloads for relational built-in functions. Spec: 4.17.9. Relational functions
- Stream fixes. Spec: 4.16. Stream class
- Device_has attribute improvements. Spec: 5.8. Attributes for device code
- Legacy type aliases. Spec: 4.14.2.2. Aliases
- Max_num_sub_groups device info query. Spec: 4.11.13.2. Kernel information descriptors
- Any_device_has/all_device_have. Spec: 4.6.4.3. Device aspects
- … …

SYCL extensions
`sycl_ext_oneapi_device_global`

Extension spec: link

Finished with development of all functionality described by the spec.

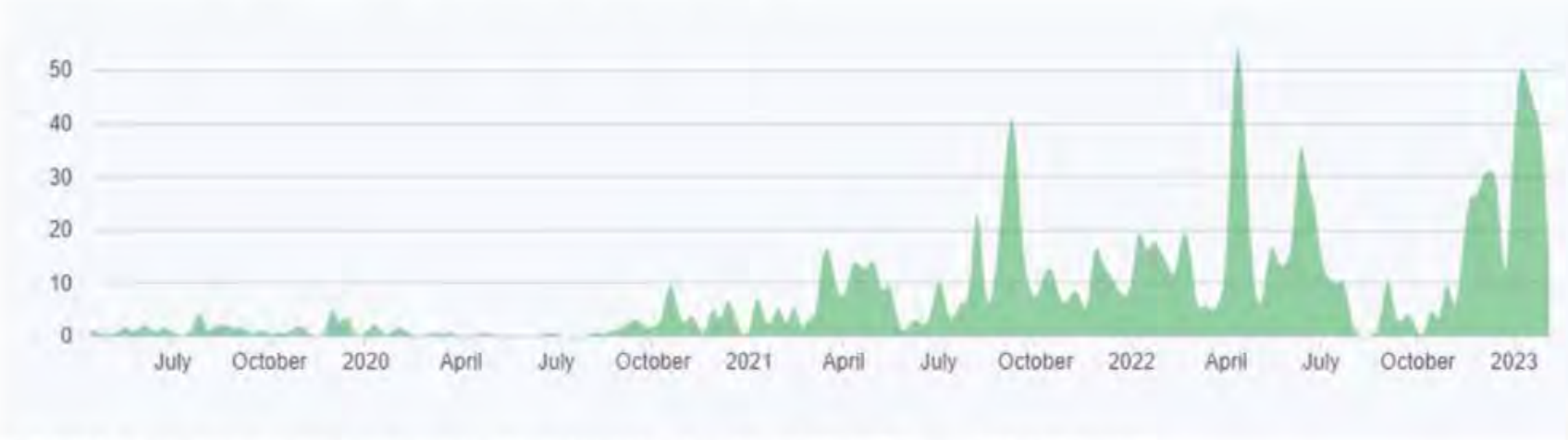
SYCL 2020 Conformance Test Suite

- Making good progress at KhronosGroup/SYCL-CTS:

Apr 14, 2019 – Feb 8, 2023

Contributions: Commits

Contributions to SYCL-2020, excluding merge commits and bot accounts



All GitHub contributions (including non-Intel contributions) to SYCL-CTS, as of 02/08/2023.

Feature Coverage:
(Not an exhaustive list)

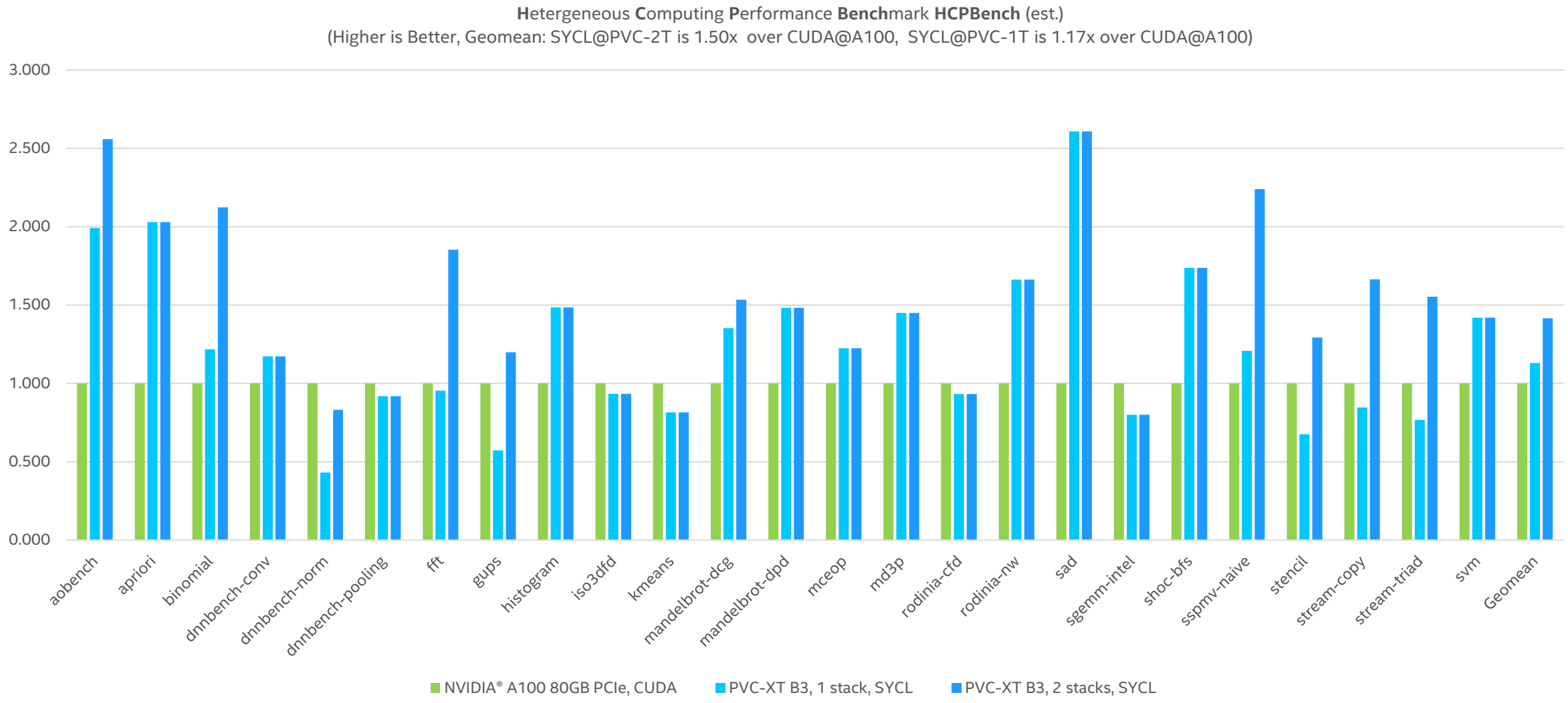
- `atomic_ref`
- Group Algorithms
- Host Tasks
- `marray`
- Reductions
- Sub-groups
- USM
- Extensions!

- oneAPI DPC++/C++ Compiler aims to pass all tests by end of 2023.

XGC (Kokkos)

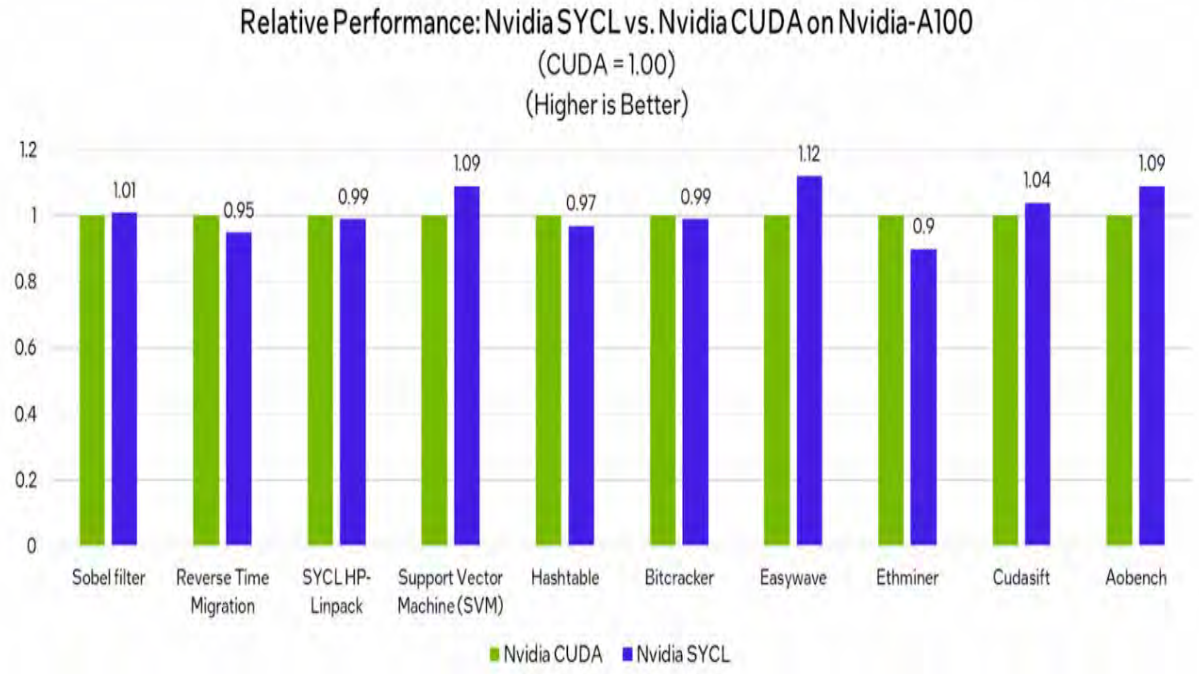
- Application and all dependent libs can be compiled / run using the oneAPI SDK
- PVC 1.54x higher FOM than provided A100 compare
- Work to resolve slow Kokkos atomics which should provide a notable speedup (currently ~20% of device time)

SYCL Workloads Performance PVC vs. A100



Performance: SYCL@A100 vs. CUDA@A100

- SYCL is getting a comparable performance to CUDA across many apps
  - Don't pay too much attention to which bar is higher or lower.
- SYCL is a high performance language for GPUs from different Vendors.
- Performance are of course also impacted by the compiler and runtime optimizations



Testing Date: Performance results are **based on testing by Intel as of August 15, 2022** and may not reflect all publicly available updates.

Configuration Details and Workload Setup: Intel® Xeon® Platinum 8360Y CPU @ 2.4GHz, 2 socket, Hyper Thread On, Turbo On, 256GB Hynix DDR4-3200, ucode 0x000363, GPU Nvidia A100 PCIe 80GB GPU memory. Software: SYCL open source/CLANG 15.0.0, CUDA SDK 11.7 with NVIDIA-NVCC 11.7.64, cuMath 11.7, cuDNN 11.7, Ubuntu 22.04.1. SYCL open source/CLANG compiler switches: -fsycl-targets=nvptx64-nvidia-cuda, NVIDIA NVCC compiler switches: -O3 -gencode arch=compute_80, code=sm_80. Represented workloads with Intel optimizations.

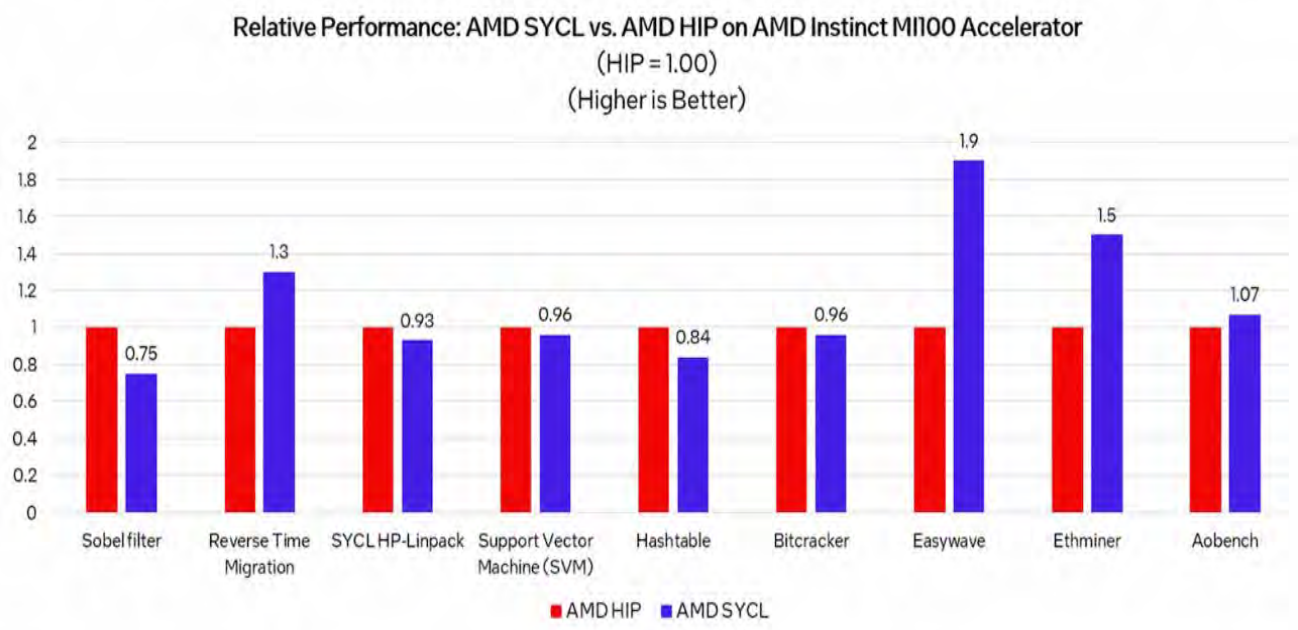
Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

Performance varies by use, configuration, and other factors. Learn more at www.Intel.com/PerformanceIndex. Your costs and results may vary.

Performance: SYCL@MI100 vs. HIP@MI100

- SYCL is getting a comparable performance to AMD HIP across many apps
  - Don't pay too much attention to which bar is higher or lower.
- SYCL is a high performance language for GPUs from different Vendors.
- Performance are of course also impacted by the compiler and runtime optimizations

Relative Performance: AMD SYCL vs. AMD HIP on AMD GPU



esting Date: Performance results are based on testing by Intel as of August 15, 2022 and may not reflect all publicly available updates.

onfiguration Details and Workload Setup: Intel® Xeon® Gold 6330 CPU @ 2.0GHz, 2 socket, Hyper Thread Off, Turbo On, 256GB Hynix DDR4-3200, ucode 0xd000363. GPU: AMD Instinct MI100, 32GB GPU memory. Software: SYCL open source/CLANG 15.0.0, AMD RoCm 5.2.1 with AMD-HIPCC 5.2.21152-4b155a06, hipSolver 5.2.1, rocBLAS 5.2.1, Ubuntu 20.04.4. SYCL open source/CLANG compiler switches: -fsycl-targets=amdgcn-amd-amdhsa -Xsycl-target-backend --offload-ch=gfx908. AMD-HIPCC compiler switches: -O3. Represented workloads with Intel optimizations.

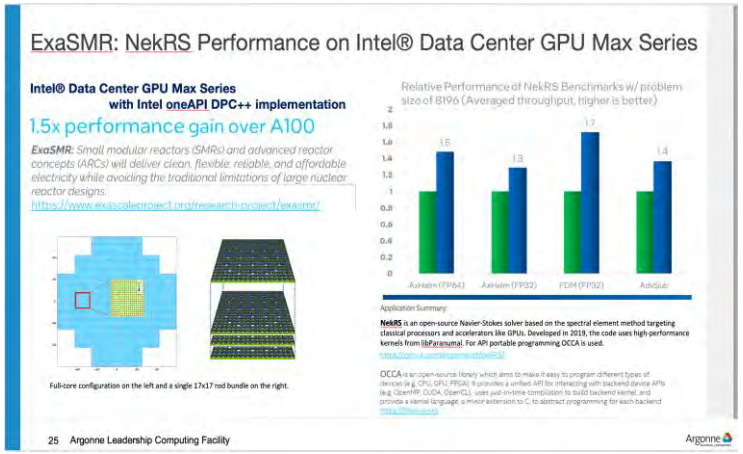
erformance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure.

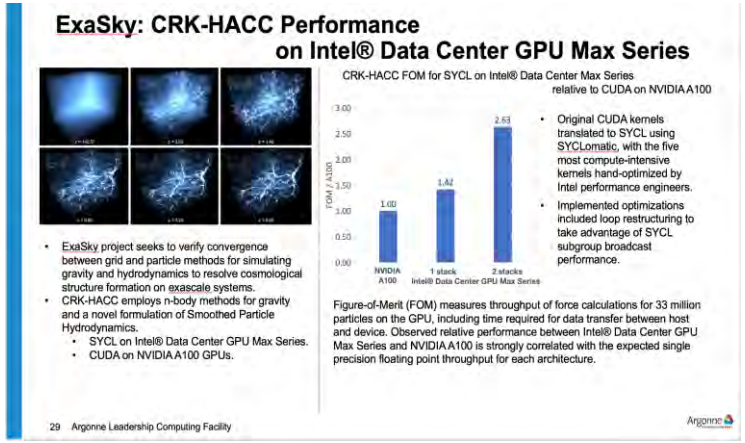
erformance varies by use, configuration, and other factors. Learn more at www.Intel.com/PerformanceIndex. Your costs and results may vary.

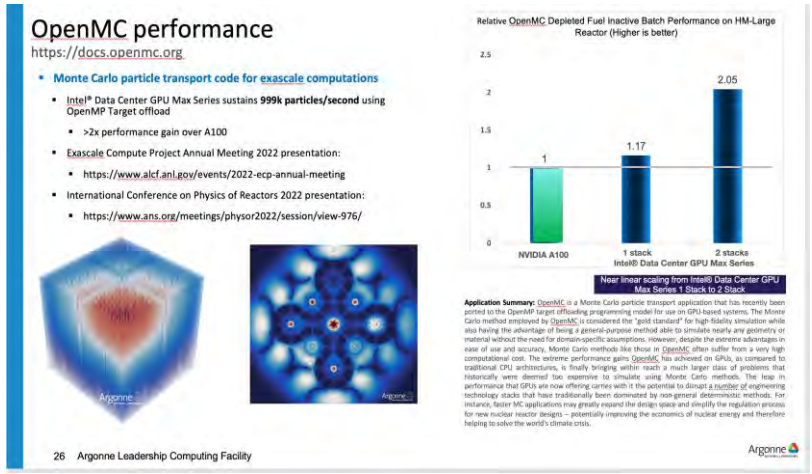
Public Presentations of Aurora Applications Results

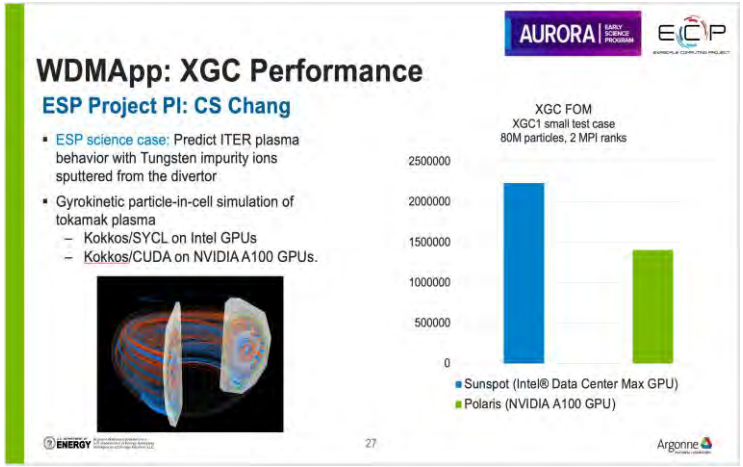
Presentations included results from 5 applications at:

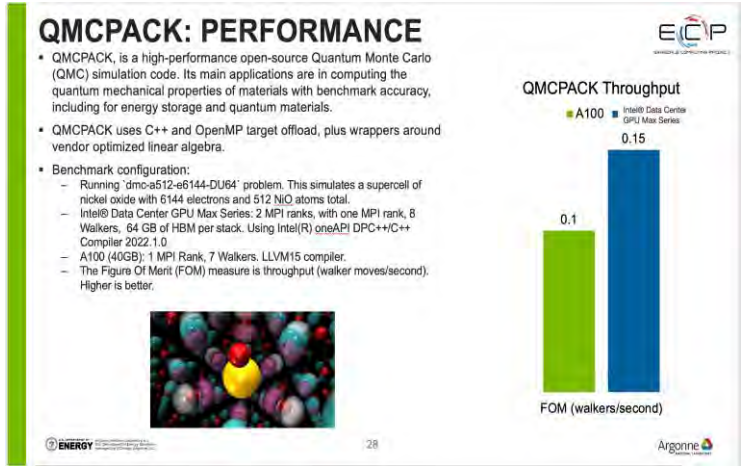
- SC'22
- HotChip'22
- One API DevSummit
- IXPUG Workshop HPC Asia 23'









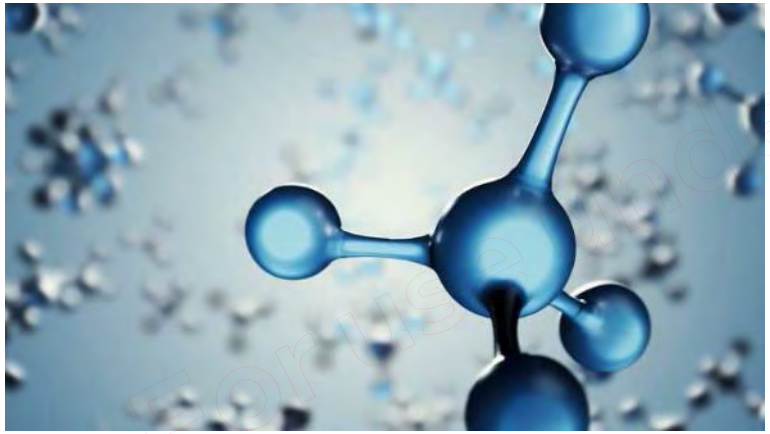


Exascale Paving the Way towards Scientific Advancement

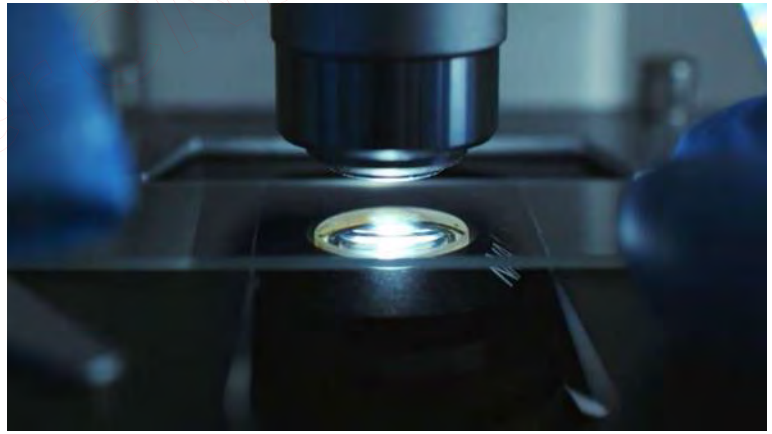
Aurora Exascale Supercomputer to Advance Clean Fusion Research

Neuroscience Research on Aurora Exascale

Exascale Computing to Power Catalysts Research



Researching Our Universe on Aurora Exascale



CANDLE Taps Deep Learning to Identify Effective Cancer Treatments

Propelling Aerospace Research on Aurora Exascale

Learn more at https://www.intel.com/content/www/us/en/high-performance-computing/supercomputing/exascale-computing.html

Summary: Key take away

- Great joint effort with Argonne Nation Lab, HPE/Cray, DOE and many partners since February 2019
- oneAPI Software Stack (SYCL, OpenMP Offloading, Compilers, Libraries, Tools) supports open standards and programming models for unlocking users from a single vendor.
- oneAPI Software Stack strives to be a vehicle for Productive, Performance and Portability.
- Aurora teams (DOE, Argonne and Intel) are in place to help all Aurora users to deliver leadership performance on Aurora system built with SPRs and PVCs.

intel®