The oneAPI DevSummit for AI and HPC 2022

Argonne NATIONAL LABORATORY

Preparing Applications for Aurora: Early Successes in Porting HPC Workloads to PVC

JaeHyuk Kwack, Scott Parker, Kris Rowe, Tim Williams, John Tramm, Thomas Applencourt, Steve Rangel, Adrian Pope, Colleen Bertoni

Argonne National Laboratory

December 7, 2022

Aurora

Leadership Computing Facility
Exascale Supercomputer

Peak Performance
≧ 2 Exaflops DP

Intel GPU
Intel® Data Center GPU Max Series

Intel Xeon Processor
4^{th} Gen Intel XEON Max Series CPU
with High Bandwidth Memory

Platform
HPE Cray-Ex

Compute Node
Two 4^{th} Gen Intel XEON Max Series CPUs
Six Intel® Data Center GPU Max Series
Node Unified Memory Architecture
Eight fabric endpoints

GPU Architecture
Intel® Data Center GPU Max Series architecture
High Bandwidth Memory Stacks

Node Performance
>130 TF

System Size
>9,000 nodes

Aggregate System Memory
>10 PB aggregate System Memory

System Interconnect
HPE Slingshot 11
Dragonfly topology with adaptive routing

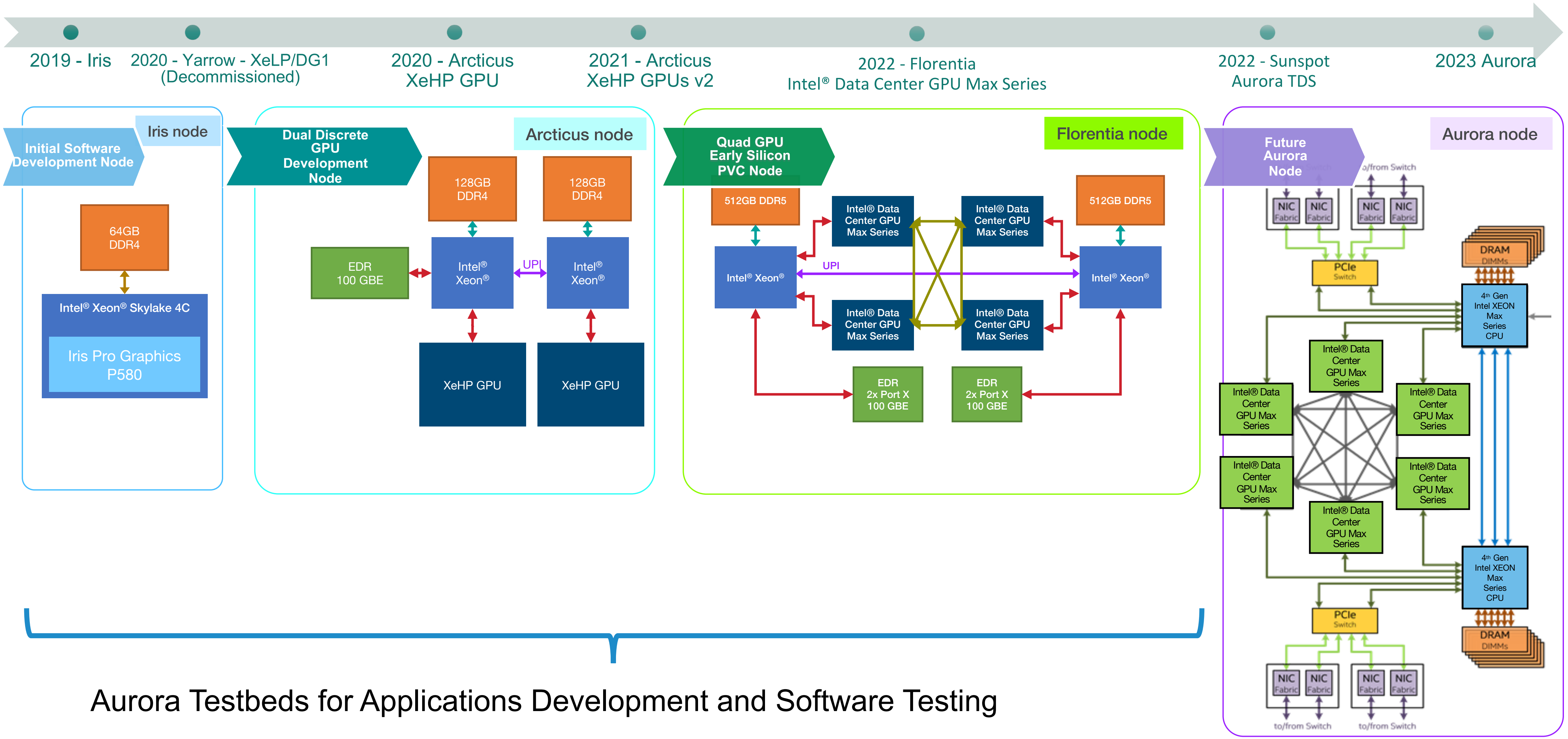
Network Switch
25.6 Tb/s per switch (64 200 Gb/s ports)
Links with 25 GB/s per direction

High-Performance Storage
220 PB
≧25 TB/s DAOS bandwidth

Software Environment
- C/C++
- Fortran
- SYCL/DPC++
- OpenMP offload
- Kokkos
- RAJA
- Intel Performance Tools

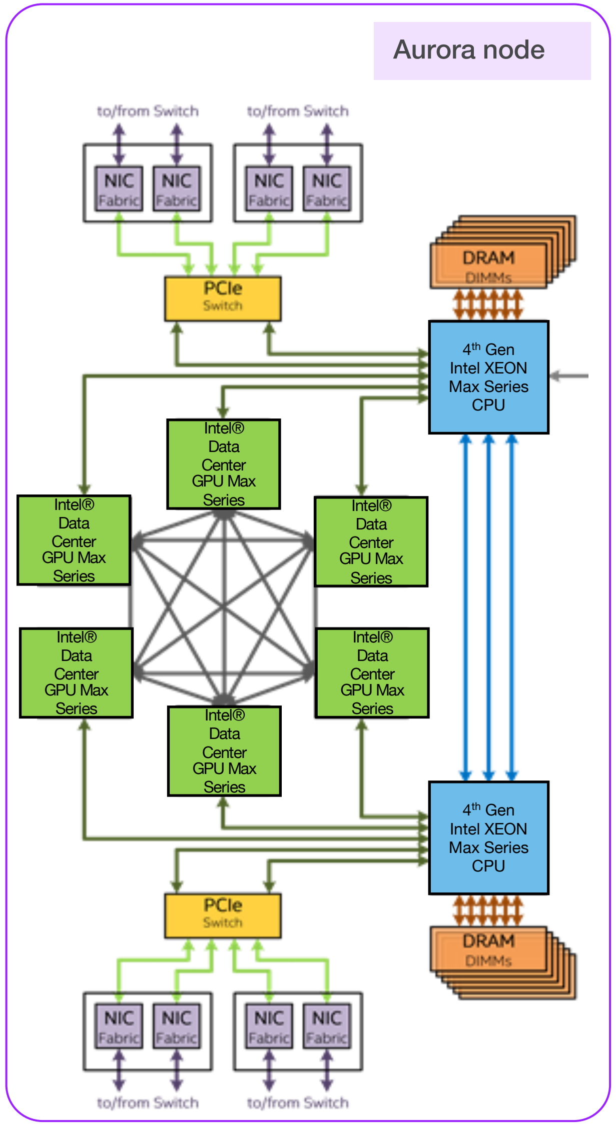
Aurora Cabinets Installed at Argonne

Aurora Testbeds to Aurora Node



Aurora Testbeds for Applications Development and Software Testing

Aurora Compute Node

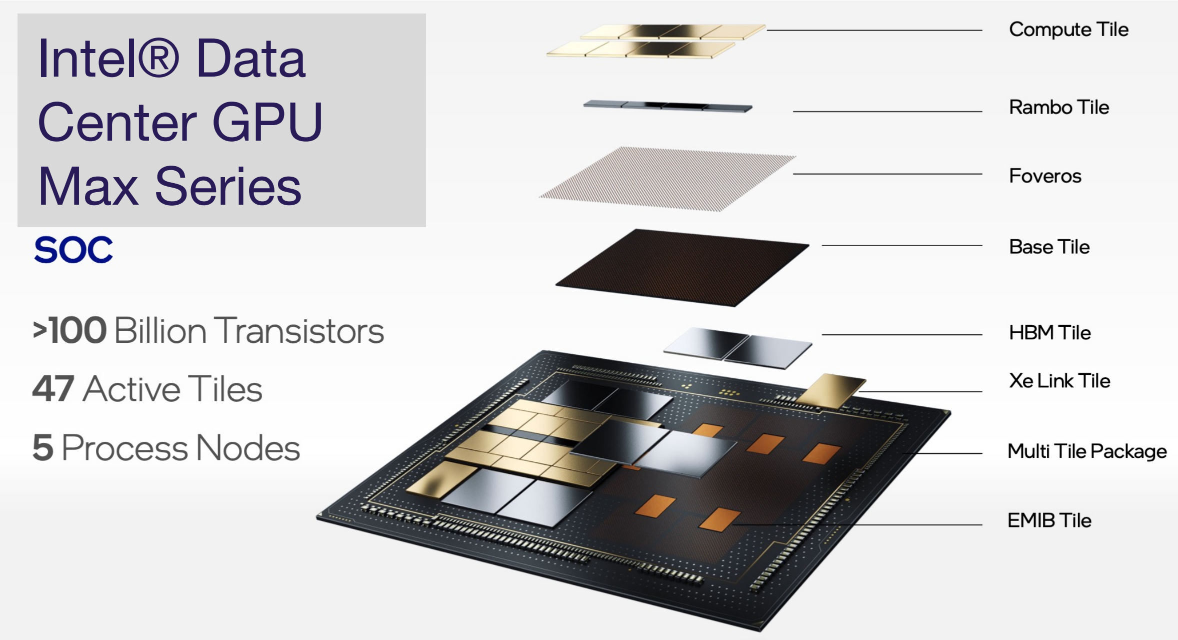


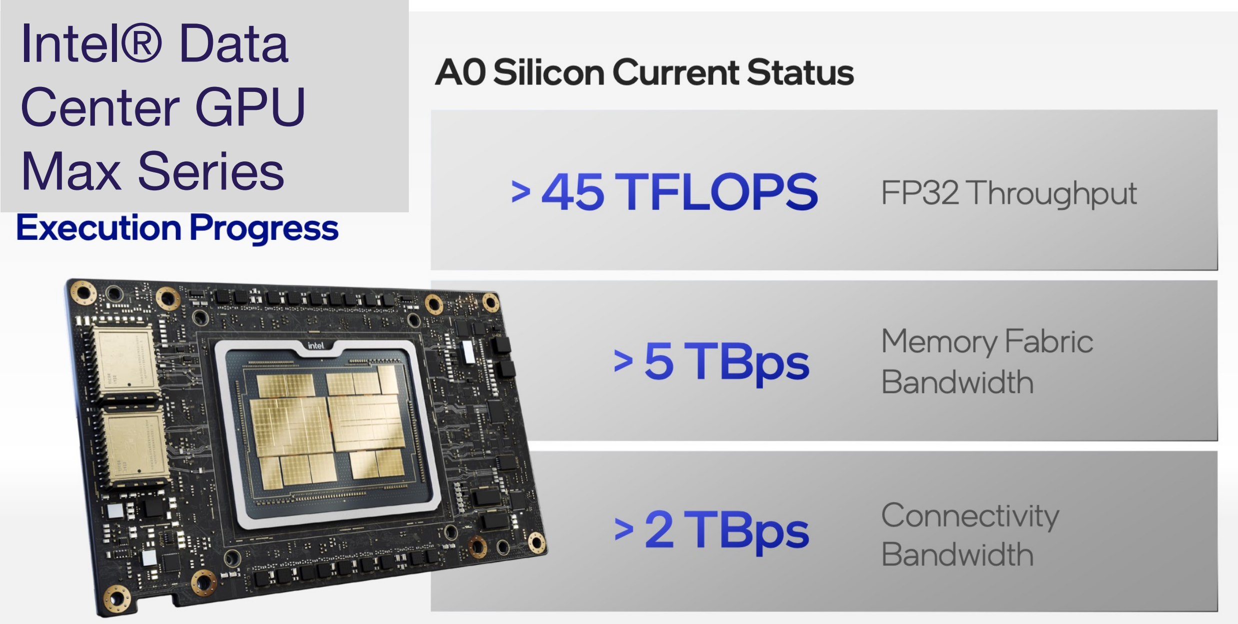
- Six Intel® Data Center GPU Max Series
  - All to all connection
- Two 4th Gen Intel XEON Max Series CPUs
- Unified Memory Architecture across CPUs and GPUs
- 8 Slingshot Fabric endpoints

Intel® Data Center GPU Max Series

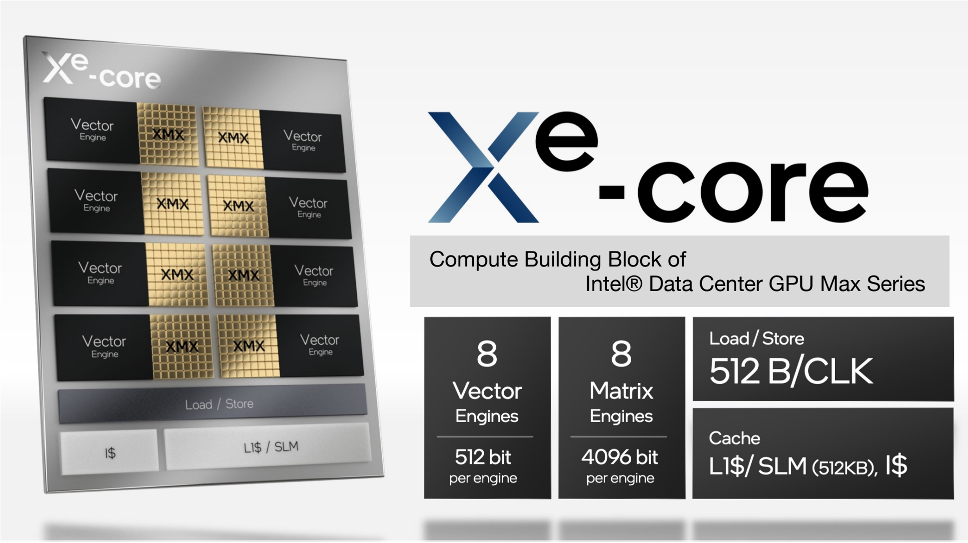
Intel provided an introduction to the Intel® Data Center GPU Max Series at their 2021 Intel Architecture Day event

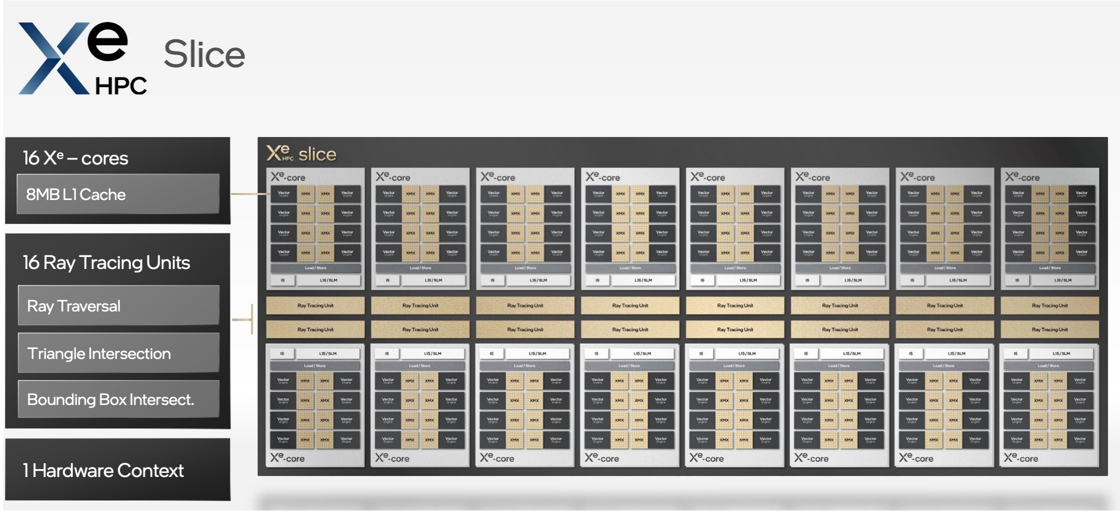
- https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html

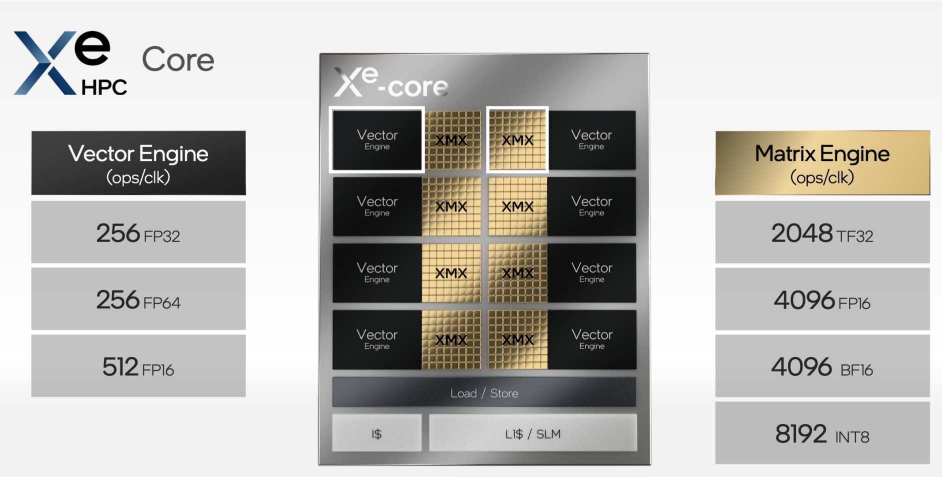


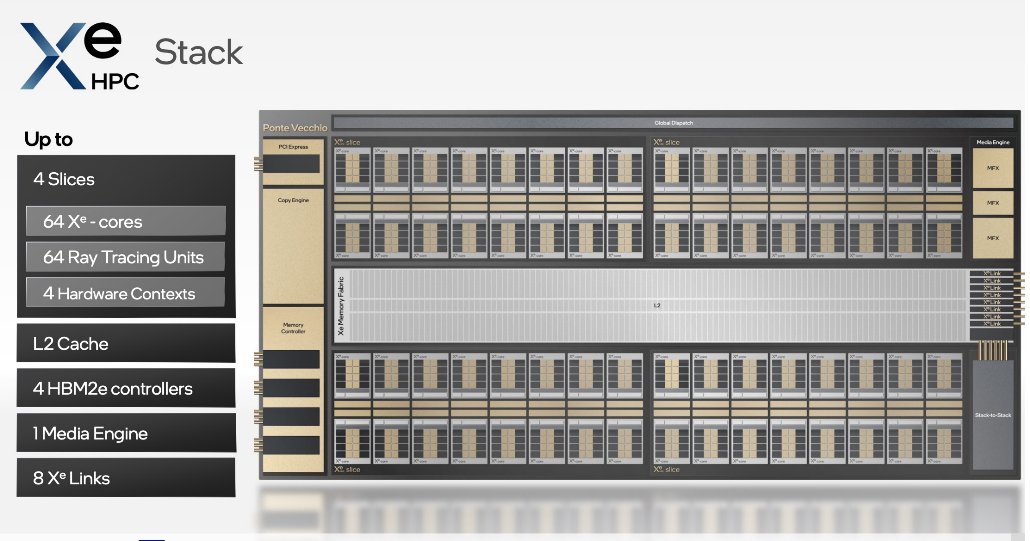


Intel® Data Center GPU Max Series Architectural Components



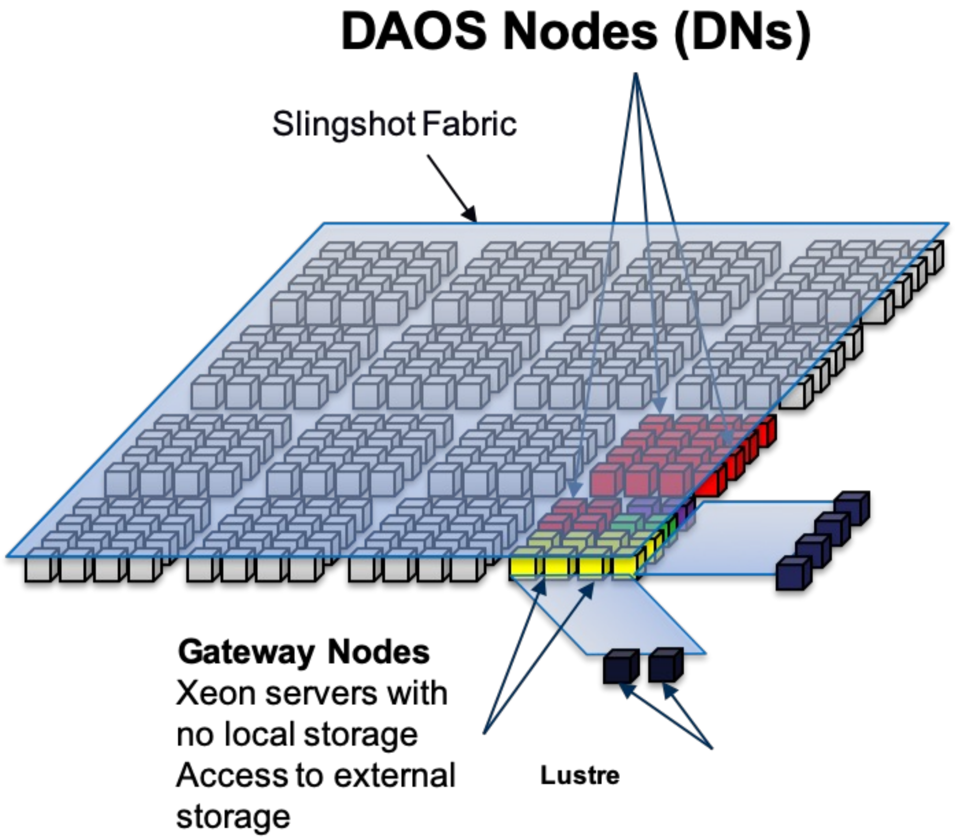






Distributed Asynchronous Object Store (DAOS)

- ❑ Primary storage system for Aurora
- ❑ Offers high performance in bandwidth and IO operations
  - ❑ 230 PB capacity
  - ❑ ≥ 25 TB/s
- ❑ Provides a flexible storage API that enables new I/O paradigms
- ❑ Provides compatibility with existing I/O models such as POSIX, MPI-IO and HDF5
- ❑ Open source storage solution



Pre-exascale and Exascale US Landscape

System	Delivery	CPU + Accelerator Vendor
Summit	2018	IBM + NVIDIA
Sierra	2018	IBM + NVIDIA
Perlmutter	2021	AMD + NVIDIA
Frontier	2021	AMD + AMD
Polaris	2021	AMD + NVIDIA
Aurora	2022	Intel + Intel
El Capitan	2023	AMD + AMD

- Heterogenous Computing (CPU + Accelerator)
- Varying vendors

oneAPI

- Industry specification from Intel (https://www.oneapi.com/spec/)
  - Language and libraries to target programming across diverse architectures (DPC++, APIs, low level interface)
- Intel oneAPI products and toolkits (https://software.intel.com/ONEAPI)
  - Languages
    - Fortran (w/ OpenMP 5+)
    - C/C++ (w/ OpenMP 5+)
    - DPC++
    - Python
  - Libraries
    - oneAPI MKL (oneMKL)
    - oneAPI Deep Neural Network Library (oneDNN)
    - oneAPI Data Analytics Library (oneDAL)
    - MPI
  - Tools
    - Intel Advisor
    - Intel VTune
    - Intel Inspector

https://software.intel.com/oneapi

Available Aurora Programming Models

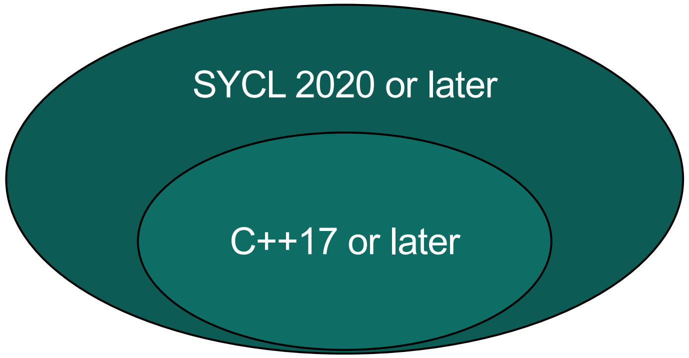
- Aurora applications may use:
  - DPC++/SYCL
  - OpenMP
  - Kokkos
  - Raja
  - OpenCL

- Experimental
  - HIP

- Not available on Aurora:
  - CUDA
  - OpenACC

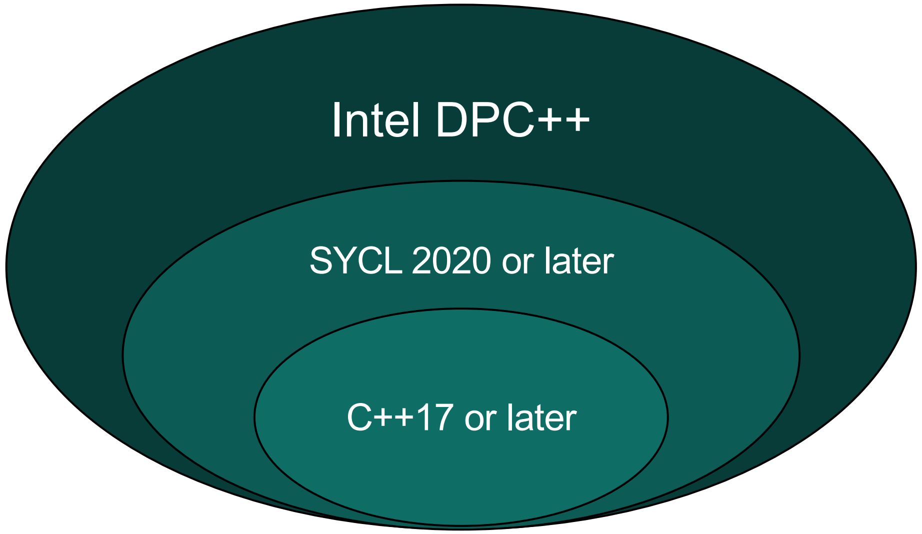
DPC++ (Data Parallel C++) and SYCL

- SYCL
  - Standard developed by Khronos and announced in 2014
  - The latest SYCL specification (SYCL 2020) was released in 2021
  - SYCL is a C++ based abstraction layer (standard C++17)
  - Builds on OpenCL **concepts** (but single-source)
  - *SYCL is designed to be as close to standard C++ as possible*



DPC++ (Data Parallel C++) and SYCL

- SYCL
  - Standard developed by Khronos and announced in 2014
  - The latest SYCL specification (SYCL 2020) was released in 2021
  - SYCL is a C++ based abstraction layer (standard C++17)
  - Builds on OpenCL **concepts** (but single-source)
  - *SYCL is designed to be as close to standard C++ as possible*



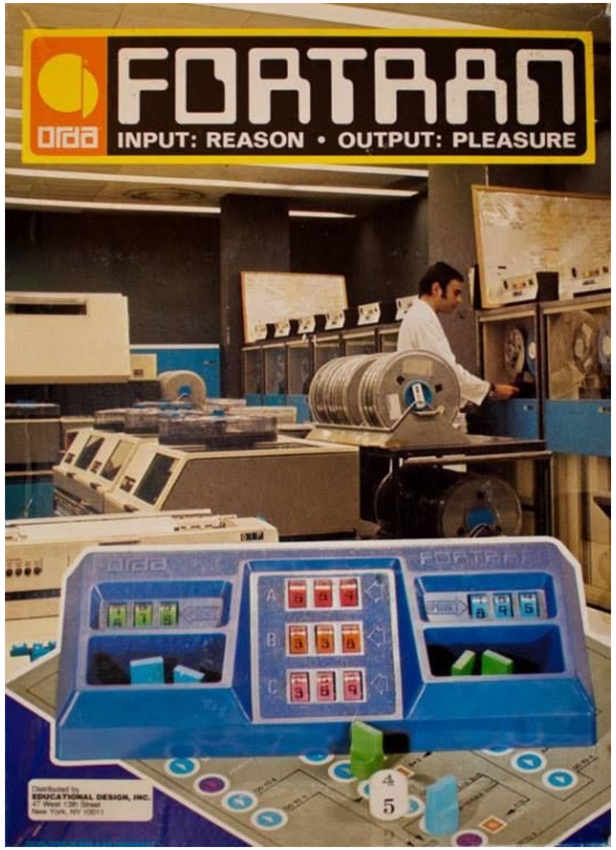
- DPC++
  - Part of Intel oneAPI specification and Intel's implementation of SYCL
  - Intel extension of SYCL to support new innovative features
  - Open source and available on GitHub
  - Contains a Plugin Interface (PI) to allow DPC++ to run on multiple devices

OpenMP

- OpenMP is a widely supported and utilized programming model
- OpenMP 5 constructs will provide directives based programming model for Intel GPUs
- Available for C, C++, and Fortran and optimized for Aurora
- Current OpenMP 5.1 spec supports offloading to an accelerator/GPU
  - Support started with OpenMP 4
- OpenMP with offload support offers a potential path to developing performance portable applications
- Multiple compilers and vendors providing OpenMP implementations
- Community has a consensus what is the “most common” subset of OpenMP features to be supported on devices.
  - OpenMP features inappropriate to GPUs are often not implemented

Intel Fortran for Aurora

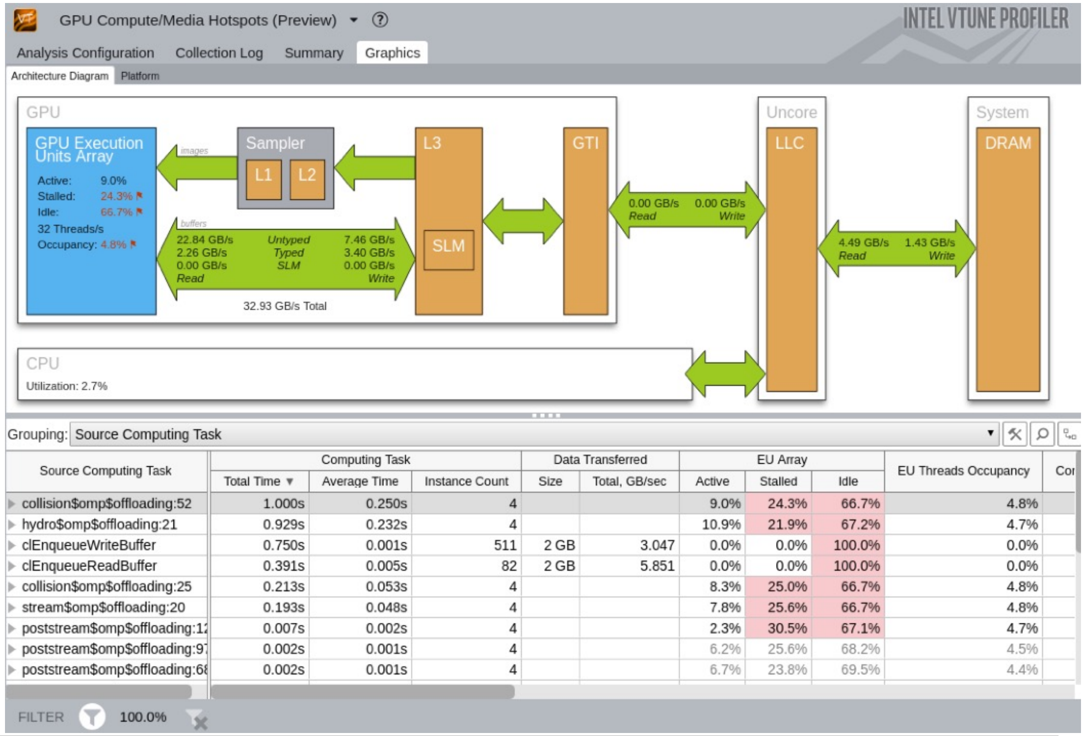
- ❑ Fortran 2008
- ❑ OpenMP 5
- ❑ New compiler—LLVM backend
  - ❑ Strong Intel history of optimizing Fortran compilers
- ❑ Beta available today in oneAPI toolkits

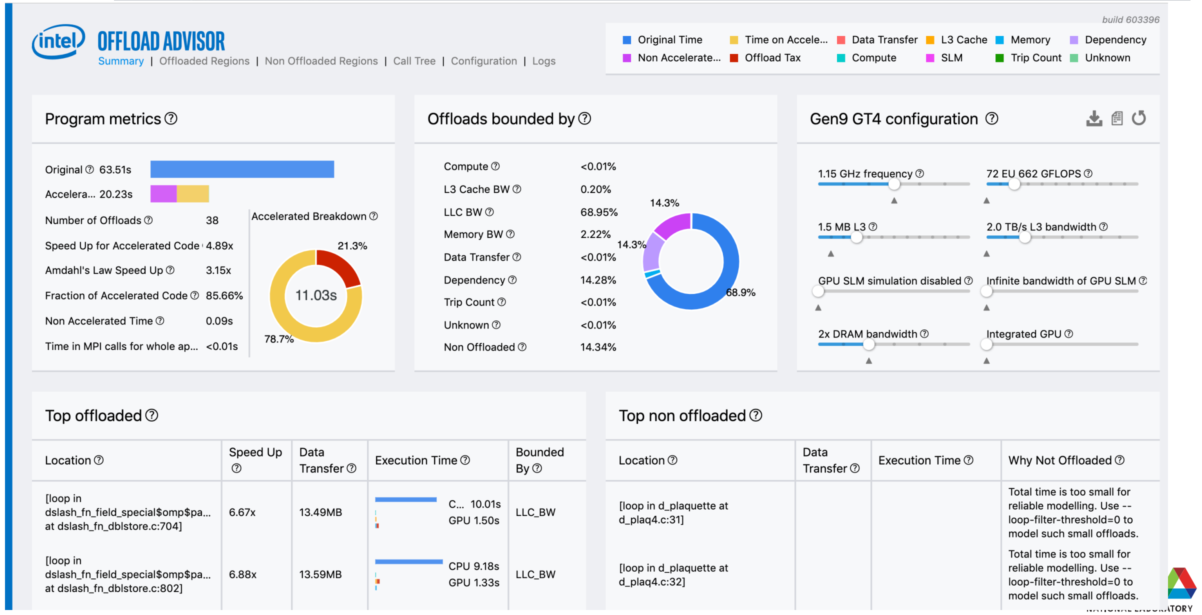


https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/fortran-compiler.html

Intel VTune and Advisor

- Vtune Profiler
  - Widely used performance analysis tool
  - Supports analysis on Intel GPUs
- Advisor
  - Provides roofline analysis
  - Offload analysis will identify components for profitable offload
    - Measure performance and behavior of original code
    - Model specific accelerator performance to determine offload opportunities
    - Considers overhead from data transfer and kernel launch





Intel MKL – Math Kernel Library

- Highly tuned algorithms
  - FFT
  - Linear algebra (BLAS, LAPACK)
  - Sparse linear algebra
  - Statistical functions
  - Vector math
  - Random number generators

- Optimized for every Intel platform

- oneAPI MKL (oneMKL)
  - https://software.intel.com/en-us/oneapi/mkl

Latest oneAPI toolkits include DPC++ support and C/Fortran OpenMP offload

AI and Analytics

- Libraries to support AI and Analytics
  - oneAPI Deep Neural Network Library (oneDNN)
    - High Performance Primitives to accelerate deep learning frameworks
    - Powers TensorFlow, PyTorch, MXNet, Intel Caffe, and more
  - oneAPI Data Analytics Library (oneDAL)
    - Classical Machine Learning Algorithms
    - Easy to use one-line daal4py Python interfaces
    - Powers Scikit-learn
  - Apache Spark MLlib

Intel® oneAPI Tools for HPC

Intel® oneAPI HPC Toolkit

Deliver Fast Applications that Scale

What is it?

A toolkit that adds to the Intel® oneAPI Base Toolkit for building high-performance, scalable parallel code on C++, Fortran, OpenMP & MPI from enterprise to cloud, and HPC to AI applications.

Who needs this product?

- OEMs/ISVs
- C++, Fortran, OpenMP, MPI Developers

Why is this important?

- Accelerate performance on Intel® Xeon® and Core™ Processors and Intel® Accelerators
- Deliver fast, scalable, reliable parallel code with less effort built on industry standards

Learn More: intel.com/oneAPI-HPCKit

Intel® oneAPI Base & HPC Toolkits

Direct Programming

- Intel® C++ Compiler Classic
- Intel® Fortran Compiler Classic
- Intel® Fortran Compiler (Beta)
- Intel® oneAPI DPC++/C++ Compiler
- Intel® DPC++ Compatibility Tool
- Intel® Distribution for Python
- Intel® FPGA Add-on for oneAPI Base Toolkit

API-Based Programming

- Intel® MPI Library
- Intel® oneAPI DPC++ Library oneDPL
- Intel® oneAPI Math Kernel Library - oneMKL
- Intel® oneAPI Data Analytics Library - oneDAL
- Intel® oneAPI Threading Building Blocks - oneTBB
- Intel® oneAPI Video Processing Library - oneVPL
- Intel® oneAPI Collective Communications Library oneCCL
- Intel® oneAPI Deep Neural Network Library - oneDNN
- Intel® Integrated Performance Primitives – Intel® IPP

Analysis & debug Tools

- Intel® Inspector
- Intel® Trace Analyzer & Collector
- Intel® Cluster Checker
- Intel® VTune™ Profiler
- Intel® Advisor
- Intel® Distribution for GDB

Intel® oneAPI **HPC** Toolkit +

Intel® oneAPI **Base** Toolkit

intel. 1 oneAPI HPC TOOLKIT

Aurora Applications Overview

- ALCF and Intel are working with over 40 projects to ready codes for Aurora:
  - Argonne Early Science Program (ESP) projects contains a mix of simulations, learning and data projects
  - DOE Exascale Computing Project (ECP) contains applications (AD) and software (ST) projects
- Over 50 applications and software packages are being prepared for Aurora:
- Involves effort from over 60 Argonne and Intel people and numerous outside teams
- Significant progress on readying applications for Aurora has occurred
  - ECP and ESP teams have been actively porting and testing code and reporting issues
  - Argonne and Intel have held quarterly application status reviews to identify top issues
  - Monthly priority bug meeting between ANL and Intel to follow-up and track issue resolution
  - Receiving regular SDK updates from Intel
  - Test framework on Aurora Testbeds allows issue reproducers and applications tests to be run before software updates and nightly to identify changes

Aurora Applications Projects

ExaWind (PI: Mike Sprague)
- AMR-Wind
- Nalu-Wind
- OpenFAST

Multiphysics-ESP (PI: Amanda Randles)
- HARVEY

CFDML-ESP (PI: Kenneth Jansen)
- Data-Driven CFD

PHASTA-ESP (PI: Kenneth Jansen)
- PHASTA

E3SM (PI: Mark Taylor)
- E3SM-MMF (YALK)
- E3SM-MMF

EXAALT (PI: Danny Perez)
- LATTE
- LAMMPS

LQCDML-ESP (PI: William Detmold)
- Flow-based generative model
- Multigrid parameter optimization

ExaFEL (PI: Amedeo Perazzo)
- spiniFEL
- cctbx

ExaSMR (PI: Steven Hamilton)
- OpenMC
- NekRS

ExaStar (PI: Daniel Kasen)
- FLASH-X
- Thornado

MatML-ESP (PI: Noa Marom)
- FHI-aims
- MatML Workflow
- BerkelyGW

UINTAH-ESP (PI: Martin Berzins)
- Uintah

WDMApp (PI: Amitava Bhattacharjee)
- GEM
- GENE
- XGC

XGC-ESP (PI: C.S. Chang)
- XGC

Connectomics-ESP (PI: Nicola Ferrier)
- mb_aligner
- Flood Fill Network

DarkSkyML-ESP (PI: Salman Habib)
- DarkSkyMining

ExaSky (PI: Salman Habib)
- NYX
- HACC

HACC-ESP (PI: Katrin Heitmann)
- HACC

LatticeQCD-ESP (PI: Norman Christ) , LatticeQCD (PI: Andreas Kronfeld)
- MILC
- Grid
- QUDA
- Chroma

NAQMC_RMD-ESP (PI: Aiichiro Nakano)
- RMD
- QXMD/DCMesh

ATLAS-ESP (PI: Walter Hopkins)
- MadGraph
- FastCaloSim

Catalysis-ESP, NWChemEX (PI: Theresa Windus)
- NWChemEX

Candle-ESP, Candle (PI: Rick Stevens)
- Uno

QMCPACK-ESP (PI: Anouar Benali) , QMCPACK (PI: Paul Kent)
- QMCPACK

NAMD-ESP (PI: Benoit Roux)
- NAMD

GAMESS (PI: Mark Gordon)
- GAMESS

FusionDL-ESP (PI: Williams Tang)
- FusionDL

EQSim (PI: David McCallen)
- ESSi
- SW4

MFIX-Exa (PI: Madhava Syamlal)
- MFIX-Exa

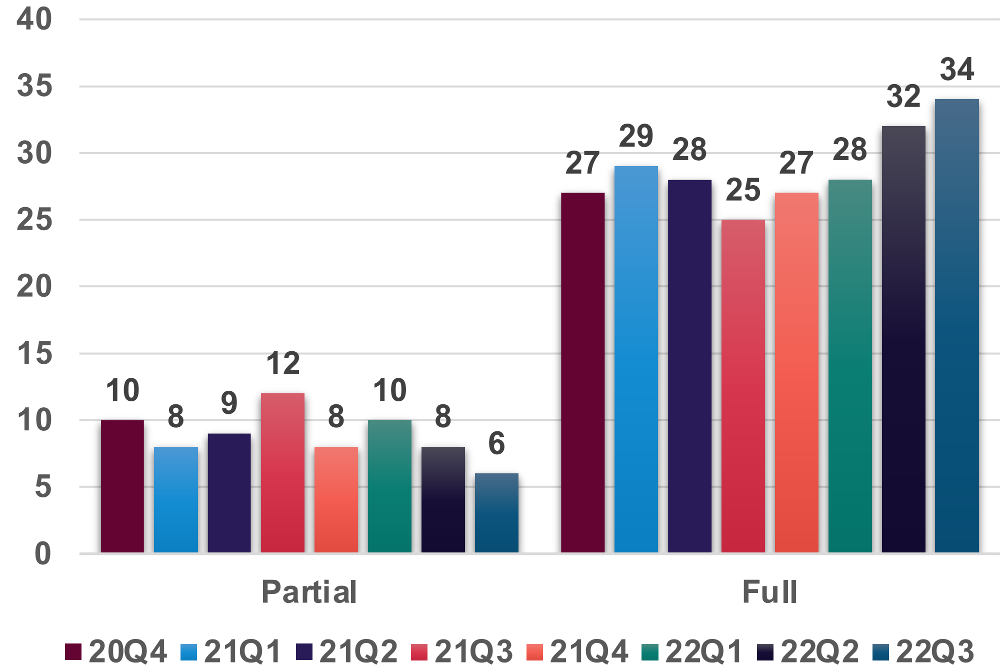
Software and Library Engagements

Kokkos/RAJA
MPICH: ExaScale MPI
HPCToolkit
ExaPAPI++
TAU
PETSc/TAO
STRUMPACK/SuperLU
HYPRE/Sundial
SLATE, HeFFTE, MAGMA
Trilinos
VTK-M

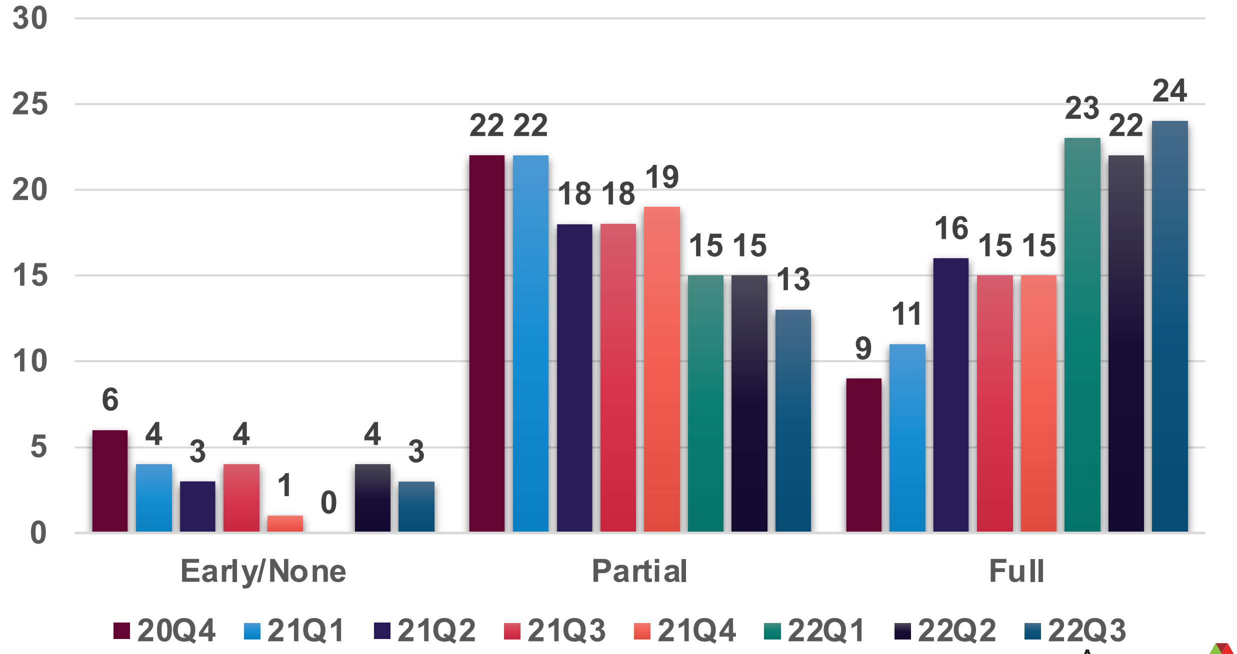
Aurora Applications Development

- **Steps in application preparation**
  - Implementation of science and algorithms
  - Porting to Aurora programming models
  - Testing with Aurora SDK on Aurora testbeds
  - Tuning for performance on Aurora testbeds
  - Scaling across the Aurora system

Application Science Implementation



Port to Aurora Programming Models



Aurora Applications Status 22Q3

Application	PVC Status
HACC	
OpenMC	
XGC	
QMCPack	
AMRWind	
NAMD	
LAMMPS	
NekRS	
QUDA	
Data Driven CFD	
SW4	
Harvey	
PHASTA	
MFIX-Exa	
FushionDL	
DCMesh	
E3SM-MMF	
CANDLE/UNO	
Thornado	
Chroma	

Application	PVC Status
GENE	
NWChemEx	
MadGraph	
FloodFillNetwork	
Grid	
GAMESS	
NYX	
BerkelyGW	
DarkSkyMining	
Uintah	
Nalu-Wind	
GEM	
mb_aligner	
RXMD-NN	
Flow Based Generative Model	
LATTE	
FastCaloSim	
spiniFEL	
cctbx	
Multi-Grid Parameter Opt.	

Running
Running
Running
Partially Running
Porting in Progress
Not Tested

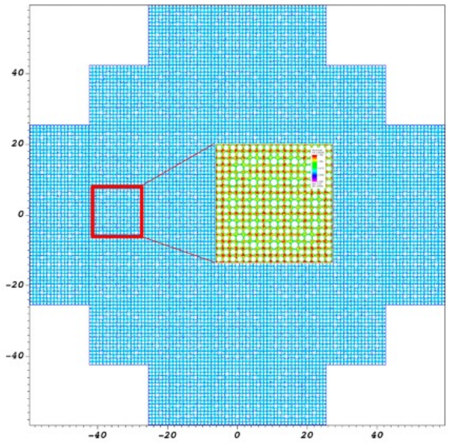
Argonne NATIONAL LABORATORY

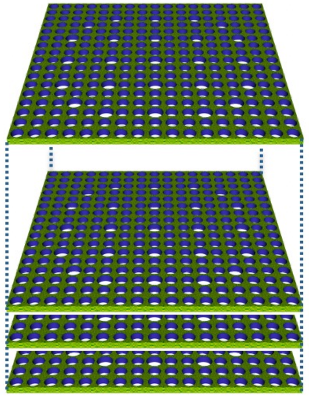
ExaSMR: NekRS Performance on Intel® Data Center GPU Max Series

Intel® Data Center GPU Max Series with Intel oneAPI DPC++ implementation

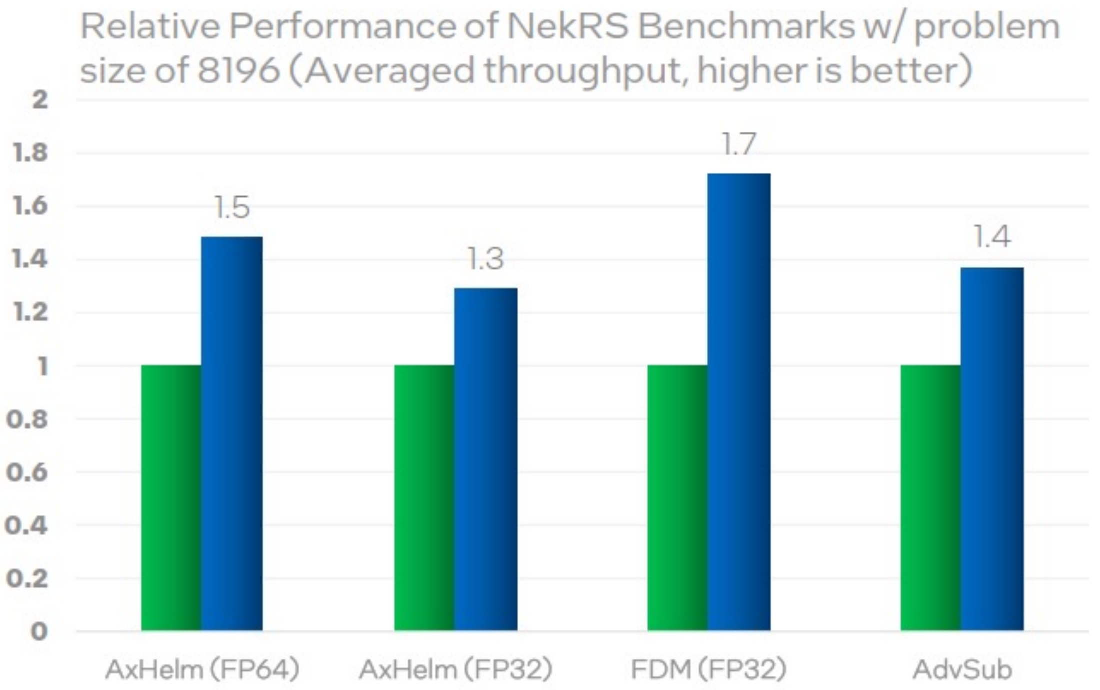
1.5x performance gain over A100

ExaSMR: *Small modular reactors (SMRs) and advanced reactor concepts (ARCs) will deliver clean, flexible, reliable, and affordable electricity while avoiding the traditional limitations of large nuclear reactor designs,*
https://www.exascaleproject.org/research-project/exasmr/





Full-core configuration on the left and a single 17x17 rod bundle on the right.



Application Summary:

NekRS is an open-source Navier-Stokes solver based on the spectral element method targeting classical processors and accelerators like GPUs. Developed in 2019, the code uses high-performance kernels from libParanumal. For API portable programming OCCA is used.

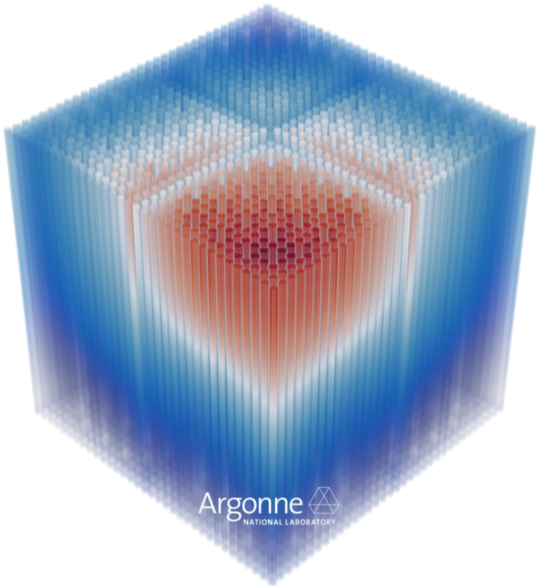
https://github.com/argonne-lcf/nekRS/

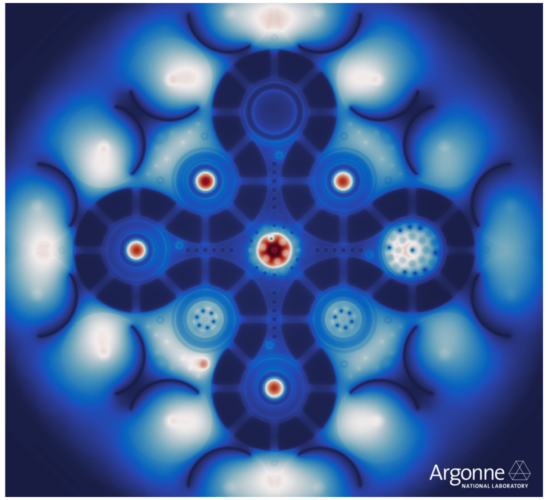
OCCA is an open-source library which aims to make it easy to program different types of devices (e.g. CPU, GPU, FPGA). It provides a unified API for interacting with backend device APIs (e.g. OpenMP, CUDA, OpenCL), uses just-in-time compilation to build backend kernel, and provide a kernel language, a minor extension to C, to abstract programming for each backend.
https://libocca.org

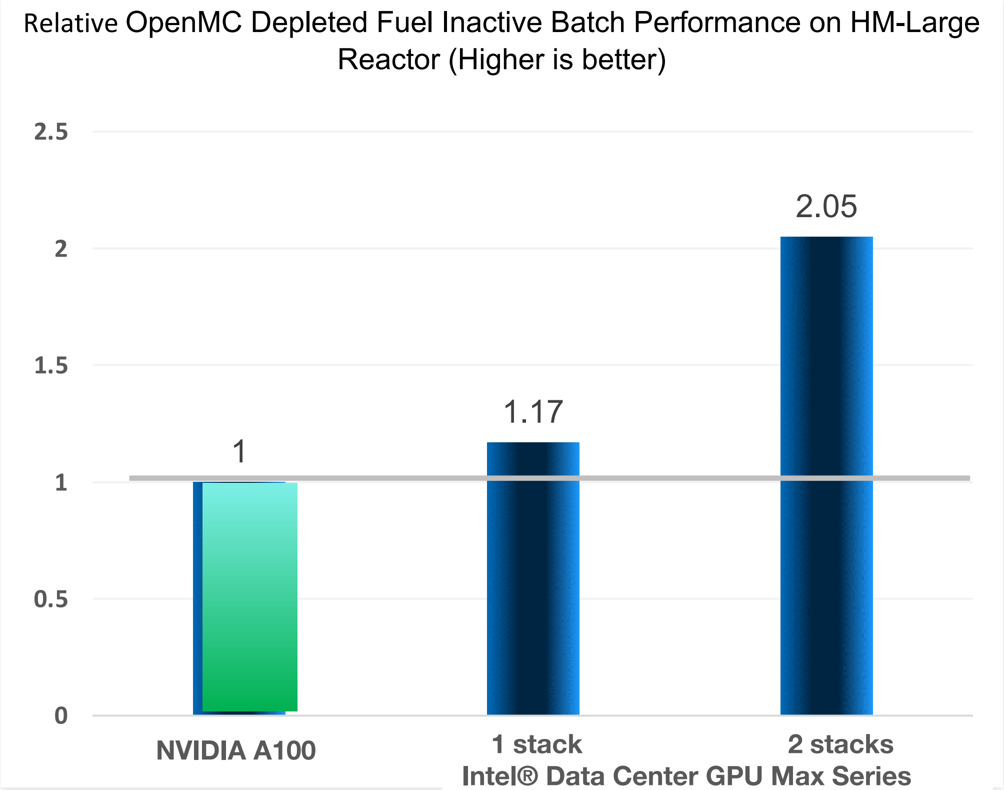
OpenMC performance

https://docs.openmc.org

- **Monte Carlo particle transport code for exascale computations**
  - Intel® Data Center GPU Max Series sustains **999k particles/second** using OpenMP Target offload
    - >2x performance gain over A100
  - Exascale Compute Project Annual Meeting 2022 presentation:
    - https://www.alcf.anl.gov/events/2022-ecp-annual-meeting
  - International Conference on Physics of Reactors 2022 presentation:
    - https://www.ans.org/meetings/physor2022/session/view-976/







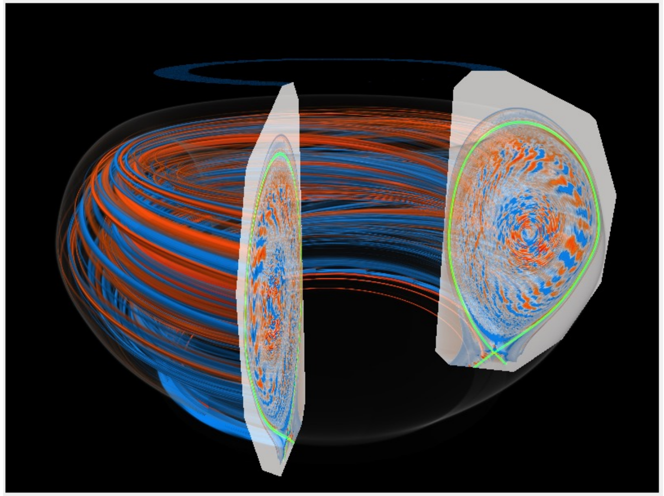
Near linear scaling from Intel® Data Center GPU Max Series 1 Stack to 2 Stack

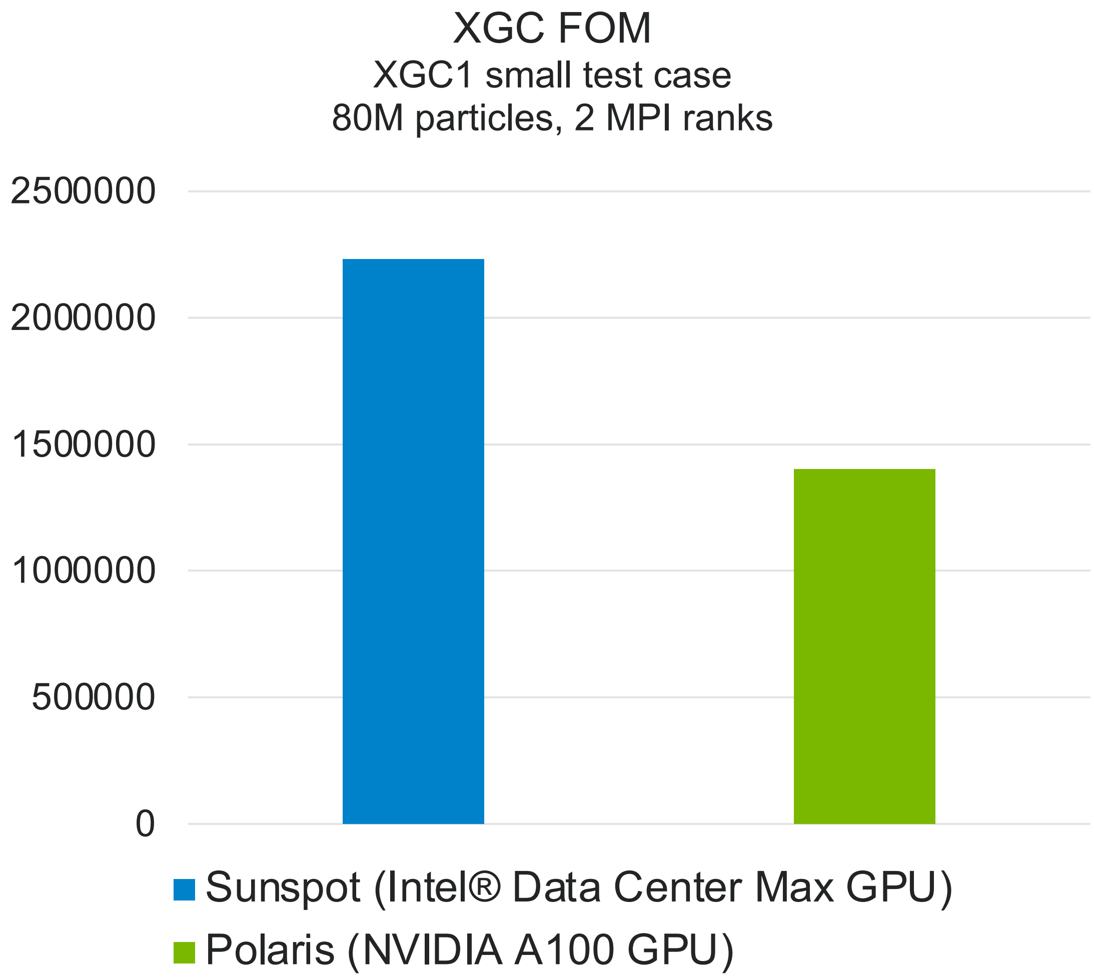
Application Summary: OpenMC is a Monte Carlo particle transport application that has recently been ported to the OpenMP target offloading programming model for use on GPU-based systems. The Monte Carlo method employed by OpenMC is considered the "gold standard" for high-fidelity simulation while also having the advantage of being a general-purpose method able to simulate nearly any geometry or material without the need for domain-specific assumptions. However, despite the extreme advantages in ease of use and accuracy, Monte Carlo methods like those in OpenMC often suffer from a very high computational cost. The extreme performance gains OpenMC has achieved on GPUs, as compared to traditional CPU architectures, is finally bringing within reach a much larger class of problems that historically were deemed too expensive to simulate using Monte Carlo methods. The leap in performance that GPUs are now offering carries with it the potential to disrupt a number of engineering technology stacks that have traditionally been dominated by non-general deterministic methods. For instance, faster MC applications may greatly expand the design space and simplify the regulation process for new nuclear reactor designs -- potentially improving the economics of nuclear energy and therefore helping to solve the world's climate crisis.

WDMApp: XGC Performance

ESP Project PI: CS Chang

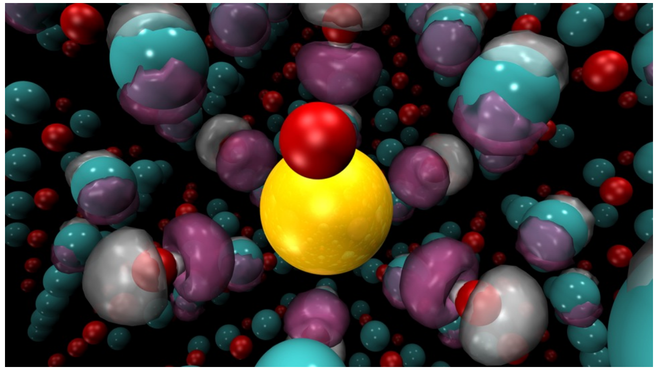
- ESP science case: Predict ITER plasma behavior with Tungsten impurity ions sputtered from the divertor
- Gyrokinetic particle-in-cell simulation of tokamak plasma
  - Kokkos/SYCL on Intel GPUs
  - Kokkos/CUDA on NVIDIA A100 GPUs.

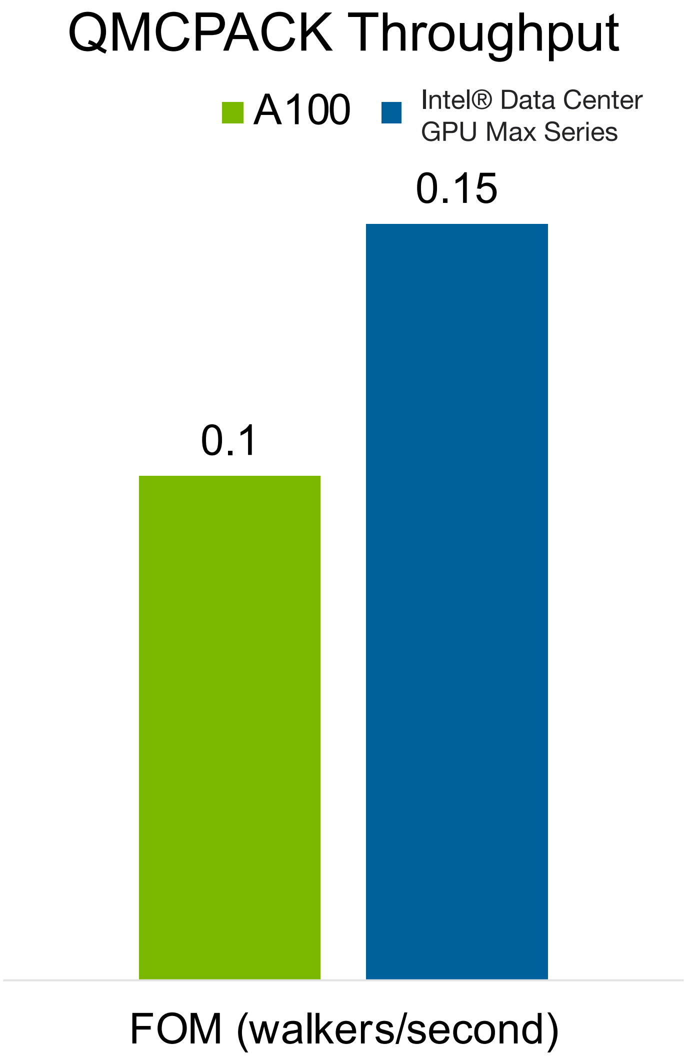




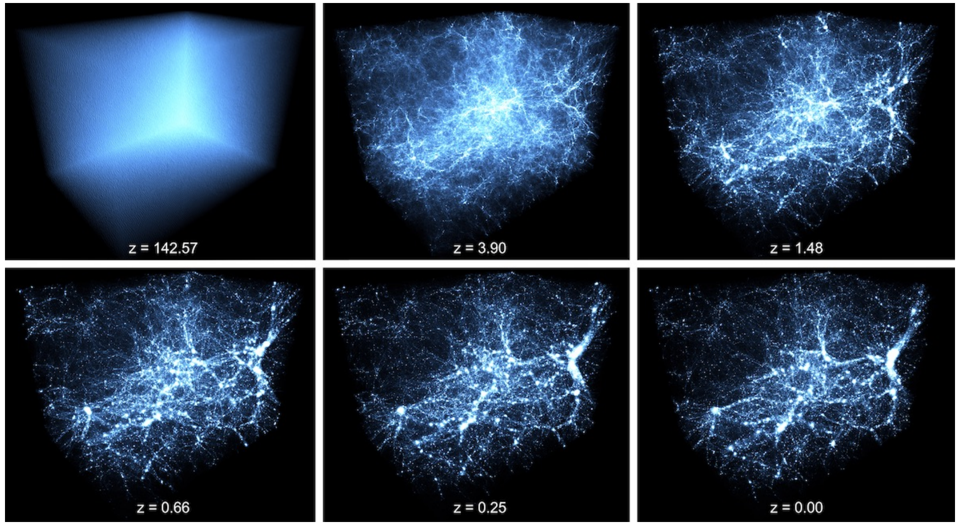
QMCPACK: PERFORMANCE

- QMCPACK, is a high-performance open-source Quantum Monte Carlo (QMC) simulation code. Its main applications are in computing the quantum mechanical properties of materials with benchmark accuracy, including for energy storage and quantum materials.
- QMCPACK uses C++ and OpenMP target offload, plus wrappers around vendor optimized linear algebra.
- Benchmark configuration:
  - Running `dmc-a512-e6144-DU64` problem. This simulates a supercell of nickel oxide with 6144 electrons and 512 NiO atoms total.
  - Intel® Data Center GPU Max Series: 2 MPI ranks, with one MPI rank, 8 Walkers, 64 GB of HBM per stack. Using Intel(R) oneAPI DPC++/C++ Compiler 2022.1.0
  - A100 (40GB): 1 MPI Rank, 7 Walkers. LLVM15 compiler.
  - The Figure Of Merit (FOM) measure is throughput (walker moves/second). Higher is better.



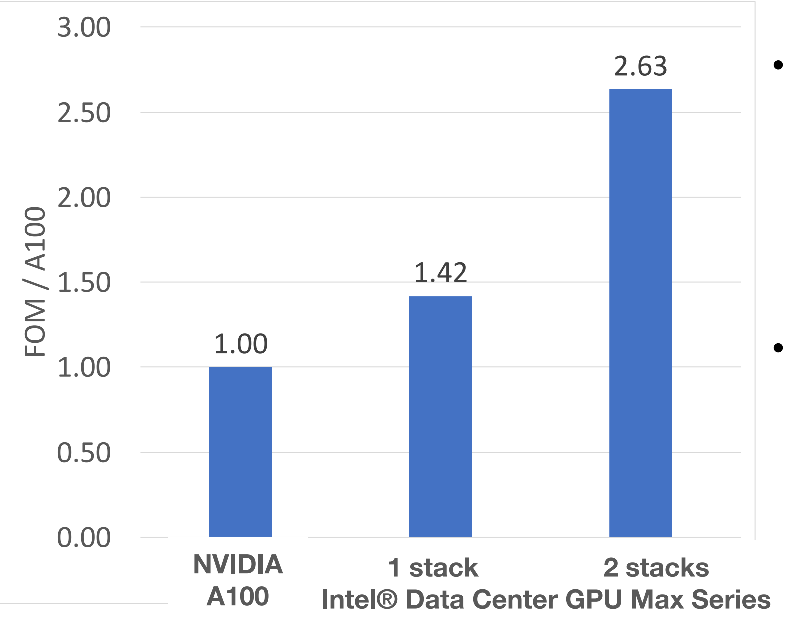


ExaSky: CRK-HACC Performance on Intel® Data Center GPU Max Series



- ExaSky project seeks to verify convergence between grid and particle methods for simulating gravity and hydrodynamics to resolve cosmological structure formation on exascale systems.
- CRK-HACC employs n-body methods for gravity and a novel formulation of Smoothed Particle Hydrodynamics.
    - SYCL on Intel® Data Center GPU Max Series.
    - CUDA on NVIDIA A100 GPUs.

CRK-HACC FOM for SYCL on Intel® Data Center Max Series relative to CUDA on NVIDIA A100



- Original CUDA kernels translated to SYCL using SYCLomatic, with the five most compute-intensive kernels hand-optimized by Intel performance engineers.
- Implemented optimizations included loop restructuring to take advantage of SYCL subgroup broadcast performance.

Figure-of-Merit (FOM) measures throughput of force calculations for 33 million particles on the GPU, including time required for data transfer between host and device. Observed relative performance between Intel® Data Center GPU Max Series and NVIDIA A100 is strongly correlated with the expected single precision floating point throughput for each architecture.

Call-to-Action

- If you are interested in Intel GPUs with Intel oneAPI toolkits for your own applications, you may try the Intel DevCloud system via the following link:
  - https://www.intel.com/content/www/us/en/developer/tools/devcloud/overview.html

Acknowledgement

- This work was supported by
  - the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357,
  - and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration).
- We also gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory.

Thank You