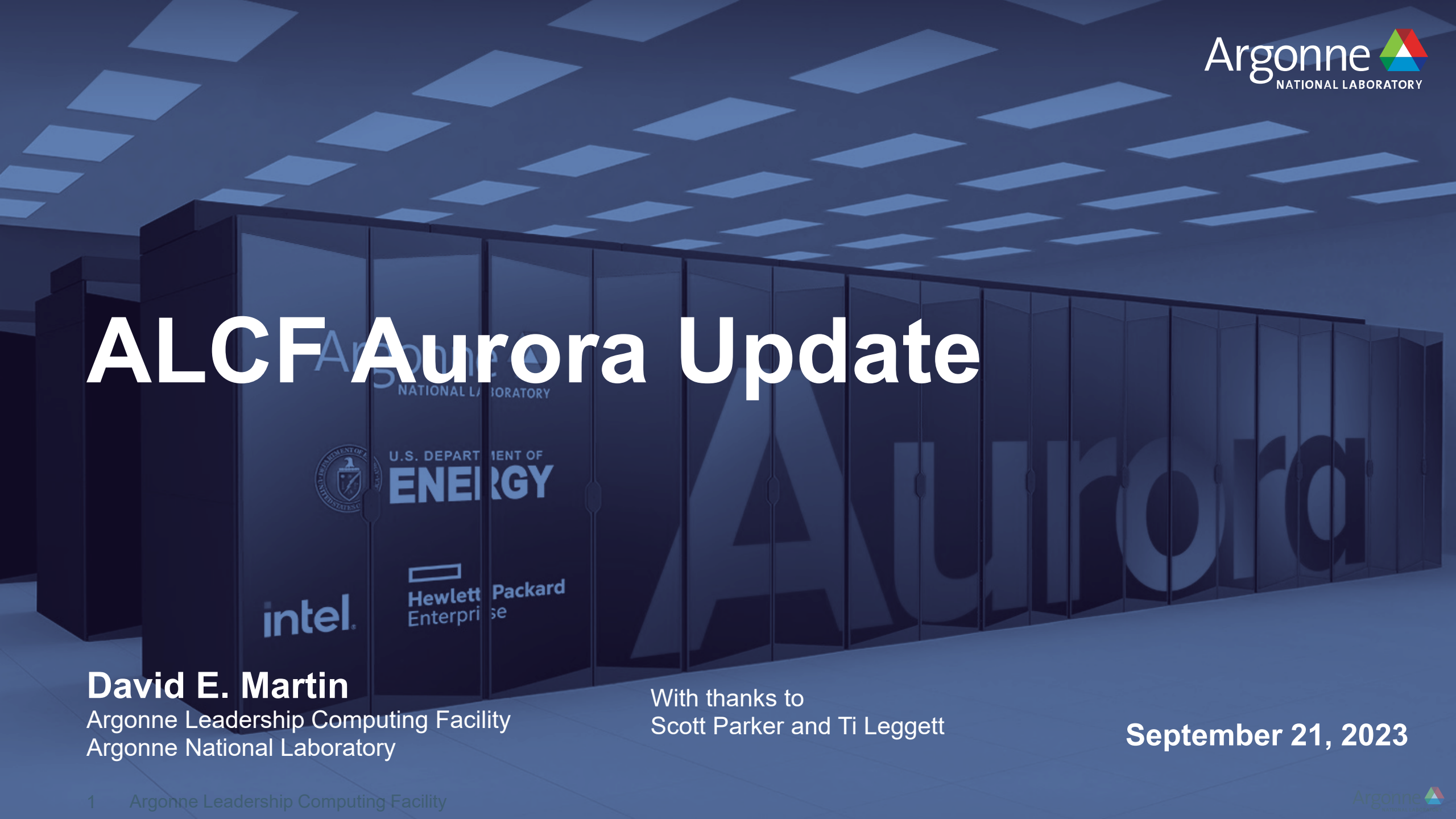


ALCF Aurora Update

David E. Martin
Argonne Leadership Computing Facility
Argonne National Laboratory

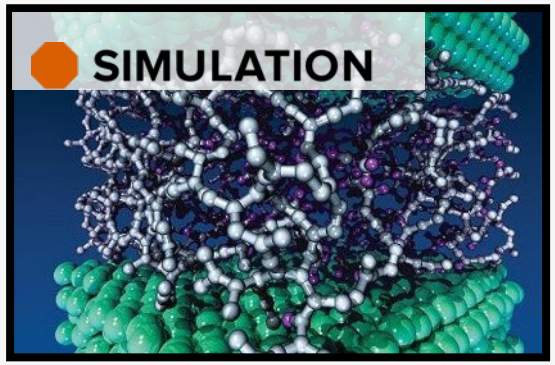
With thanks to
Scott Parker and Ti Leggett

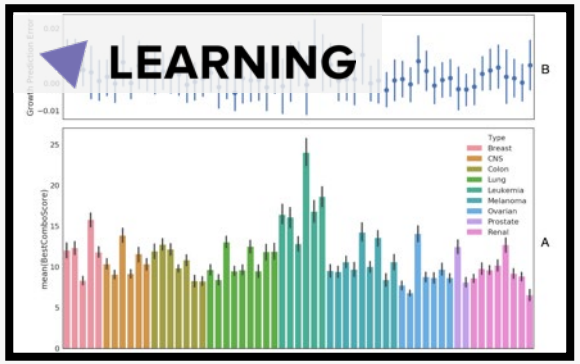
September 21, 2023

Argonne Leadership Computing Facility

The Argonne Leadership Computing Facility provides world-class computing resources to the scientific community.

- Users pursue scientific challenges
- In-house experts to help maximize results
- Resources fully dedicated to open science

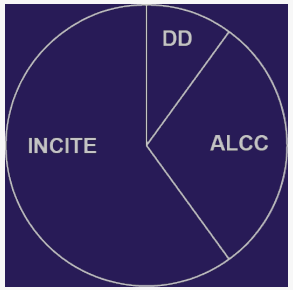




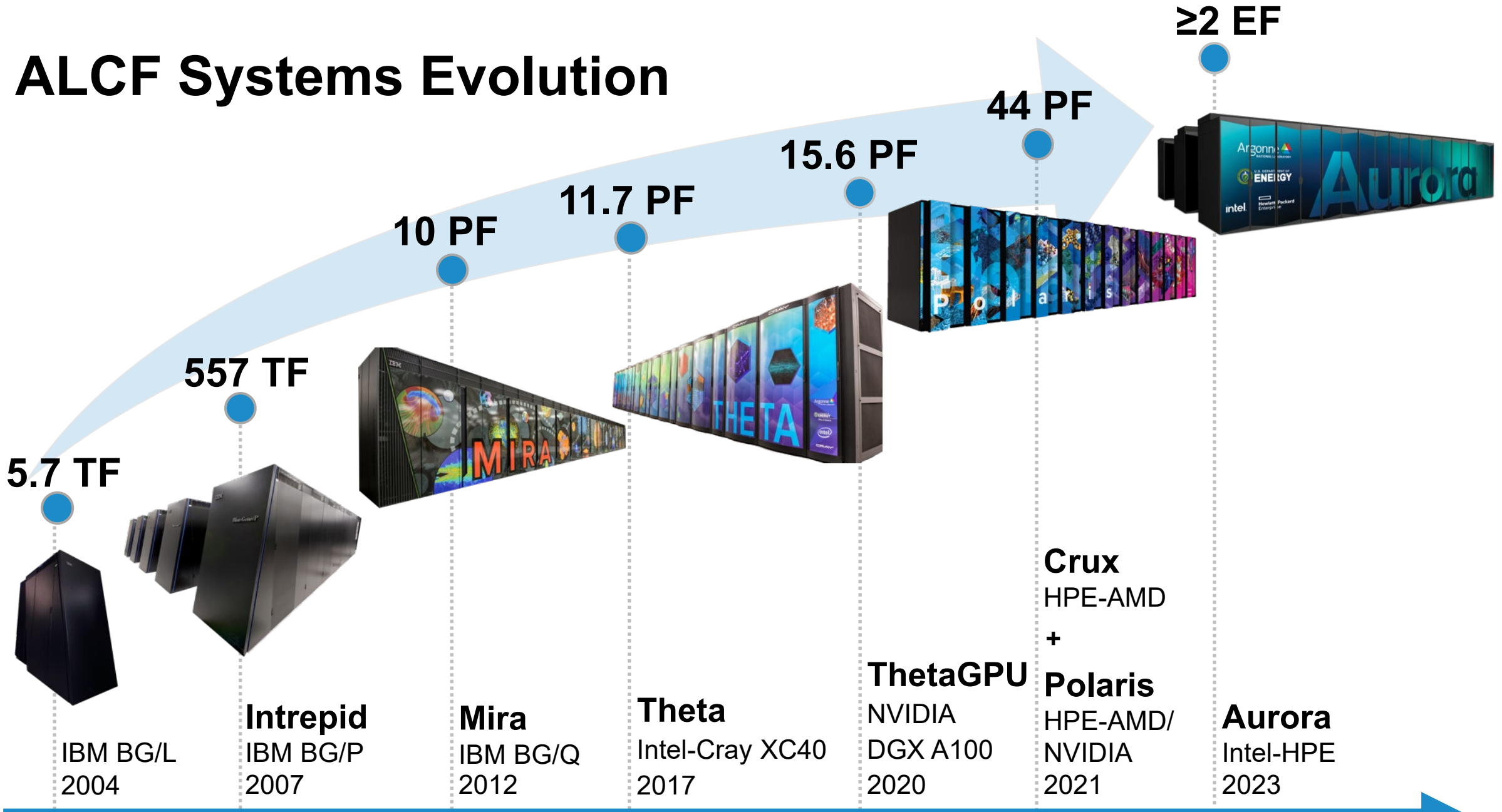
Architecture supports three types of computing

- Large-scale Simulation (PDEs, traditional HPC)
- Data Intensive Applications (scalable science pipelines)
- Deep Learning and Emerging Science AI (training and inferencing)

ALCF offers different pipelines for different applications



ALCF Systems Evolution



Intel GPU
Intel® Data Center GPU Max Series

Intel Xeon Processor
4th Gen Intel XEON Max Series CPU **with High Bandwidth Memory**

Platform
HPE Cray-Ex

Racks - 166
Nodes - 10,624
CPUs - 21,248
GPUs – 63,744

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

Platform
HPE Cray-EX

Peak FP Performance
≧ 2 Exaflops DP

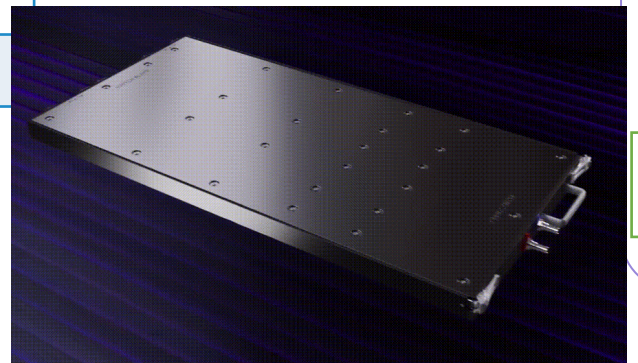
Memory
10.9PB of DDR @ 5.95 PB/s
1.36PB of CPU HBM @ 30.5 PB/s
8.16PB of GPU HBM @ 208.9 PB/s

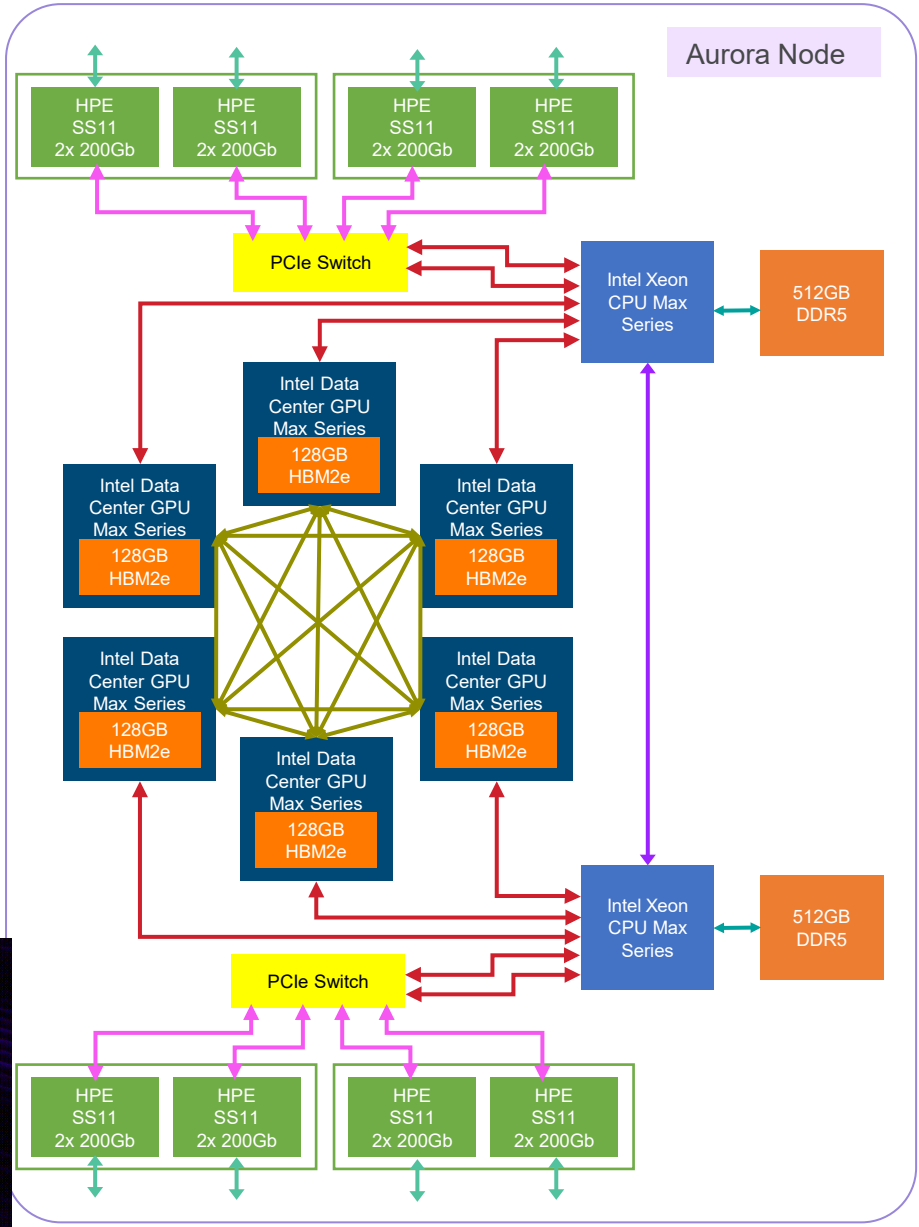
Network
2.12 PB/s Peak Injection BW
0.69 PB/s Peak Bisection BW

Storage
230PB DAOS Capacity
31 TB/s DAOS Bandwidth

Aurora Exascale Compute Blade

NODE CHARACTERISTICS
6 GPUs - Intel Data Center GPU Max Series
2 CPUs - Intel Xeon CPU Max Series
768 GB GPU HBM Memory
19.66 TB/s Peak GPU HBM BW
128 GB CPU HBM Memory
2.87 TB/s Peak CPU HBM BW
1024 GB CPU DDR5 Memory
0.56 TB/s Peak CPU DDR5 BW
≧ 130 TF Peak Node DP FLOPS
200 GB/s Max Fabric Injection
8 NICs





HPE Slingshot Interconnect

Consistent, Repeatable Application Performance

- Advanced congestion control
- Fine grained adaptive routing
- Very low average and tail latency

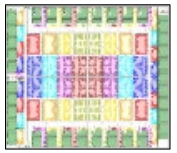
Extremely Scalable RDMA Performance

- Connectionless protocol
- Fine grained flow control
- MPI HW tag matching & progress engine
- Dragonfly topology – 3 switch hops (typical)

Native Ethernet

- Native IP – no encapsulation
- High-scale bandwidth integration to campus

HPE Slingshot Switches - 64 ports @ 200 Gbps



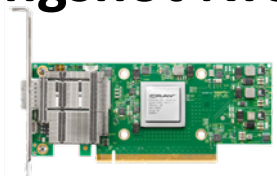
HPE Switch ASIC

Rack switches

100% DLC Switches

HPE Slingshot NICs - 200 Gbps

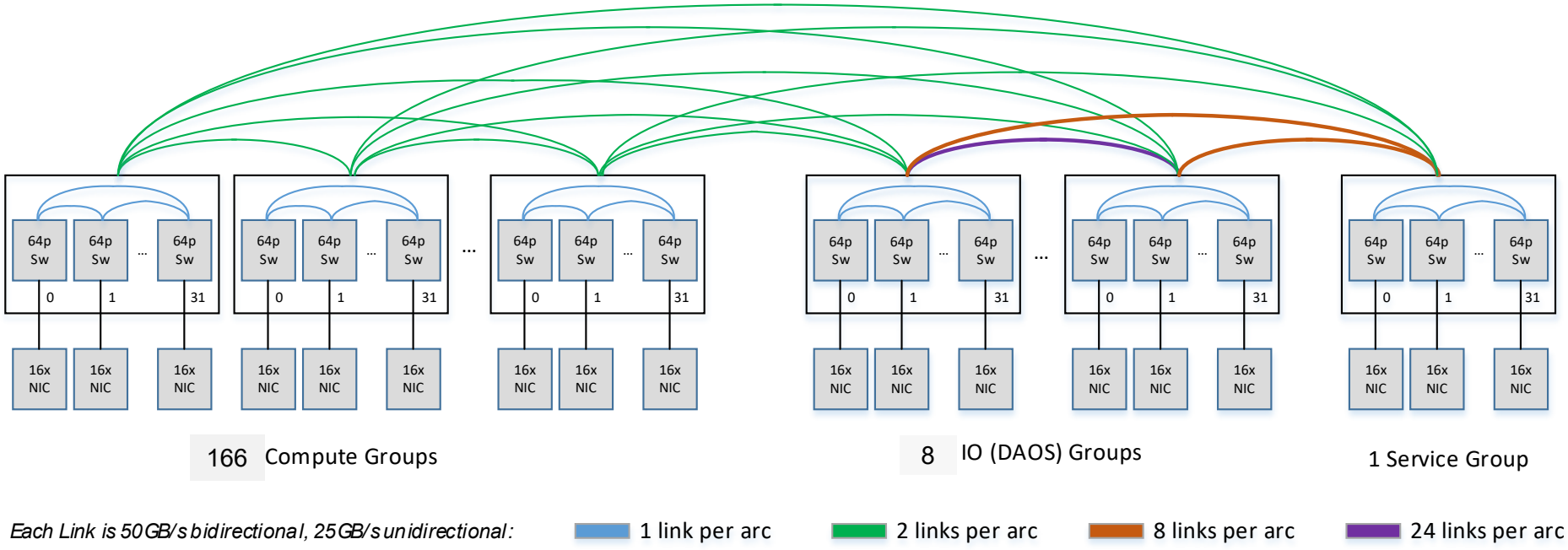
HPE NIC ASIC



PCIe Adapters

100% DLC NIC Mezz

Fabric



- 1-D Dragonfly Topology - 175 total groups (166 compute + 8 IO + 1 Service),
- All the global links are optical, all the local links in compute groups are electrical
- 2 global links between any two compute groups
- 24 links between any two IO groups, 8 links between the Service group and each IO group
- Total injection bandwidth: 2.12PB/s
- Total bisection bandwidth: 0.69PB/s

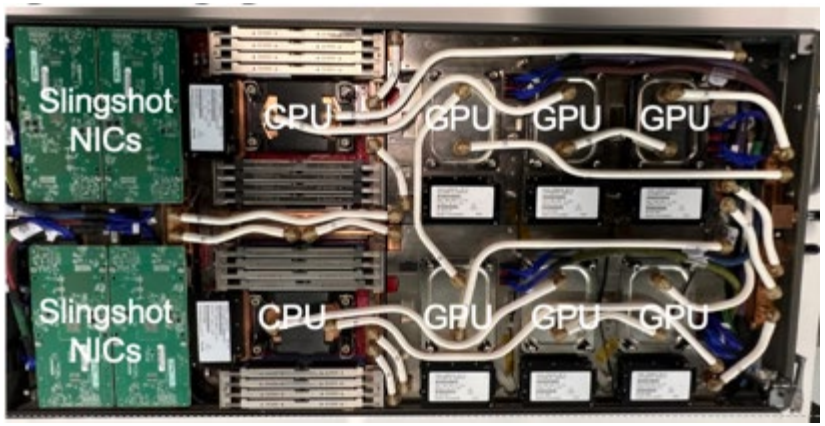
DAOS Storage Systems

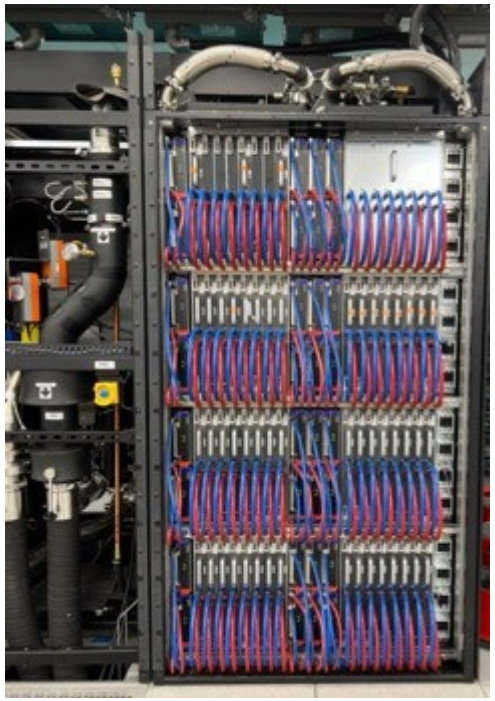
- DAOS provides Aurora's main "platform" high performance storage system
  - Provides a flexible storage API that enables new I/O paradigms
  - Provides compatibility with existing I/O models such as POSIX, MPI-IO and HDF5
  - Open source storage solution
- Aurora leverages existing Lustre storage systems, Grand and Eagle, for center-wide data access and data sharing

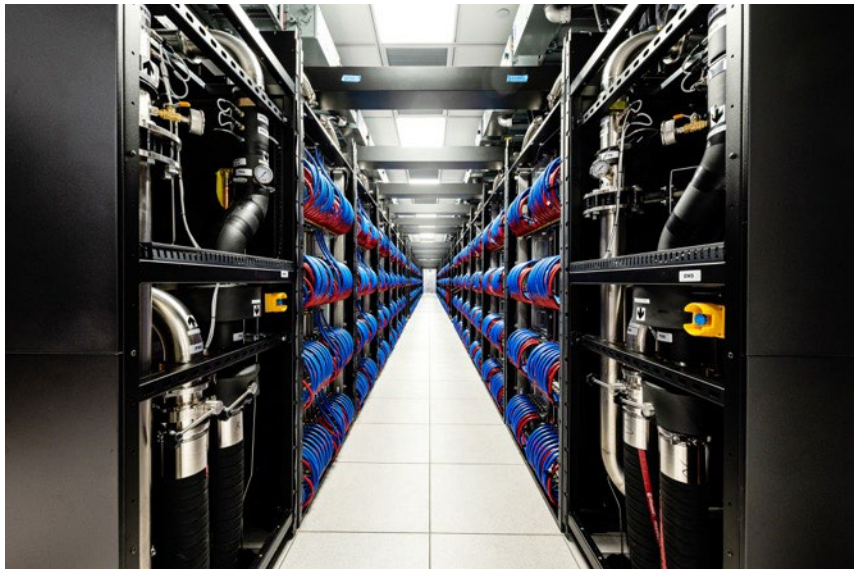
System	Capacity	Performance
Aurora DAOS	230 PB @ EC16+2 ▪ 250 PB NVMe ▪ 8 PB Optane PMEM	31 TB/s Read & Write
Eagle	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write
Grand	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write

The Status of Aurora

- All of the Aurora system is installed at ANL
  - Some compute blades still being validated
  - Fabric validation and scale up underway
- Targeting early user in Q3 2023







Aurora Applications Status at Single Node Scale

Running
Running
Running
Partially Running
Porting in Progress

Application	Status
XGC	
NAMD	
FloodFillNetwork	
HACC	
QUDA	
OpenMC	
Flash-X/Thornado	
NWChemEX	
AMR-Wind	
CANDLE/UNO	
HARVEY	
NekRS	
LAMMPS	
GENE	
FusionDL	
MadGraph	
BerkelyGW	
PHASTA	
MFIX-Exa	
Chroma	
cctbx	

Application	Status
MILC	
QMCPACK	
E3SM-MMF	
SW4	
DCMesh	
LATTE	
Grid	
GAMESS	
NYX	
Uintah	
Data Driven CFD	
DarkSkyMining	
Flow Based Generative Model	
Nalu-Wind	
GEM	
RXMD-NN	
mb_aligner	
spiniFEL	
Multi-Grid Parameter Opt.	
FastCaloSim	

Still Work To Do …

Including:

- Complete the initial system bring up and validation
- Run Early Science and ECP applications at scale
- Accept the system
- Put the system into production

But many of the major risks have been mitigated

Aurora Software

Argonne Leadership Computing Facility

Aurora oneAPI Components

Languages & Runtimes

DPC++ Compiler (CPU & GPU)

DPC++ Compatibility Tool

C/C++/Fortran OpenMP Offload Compiler (CPU & GPU)

Compiler/Compatibility IDE Plugins

Intel Distribution for Python

Parallel STL / oneDPL

oneTBB

oneCCL

Aurora MPICH

Tools

Debugger VTune Inspector

Advisor GT-PIN

Math Libraries

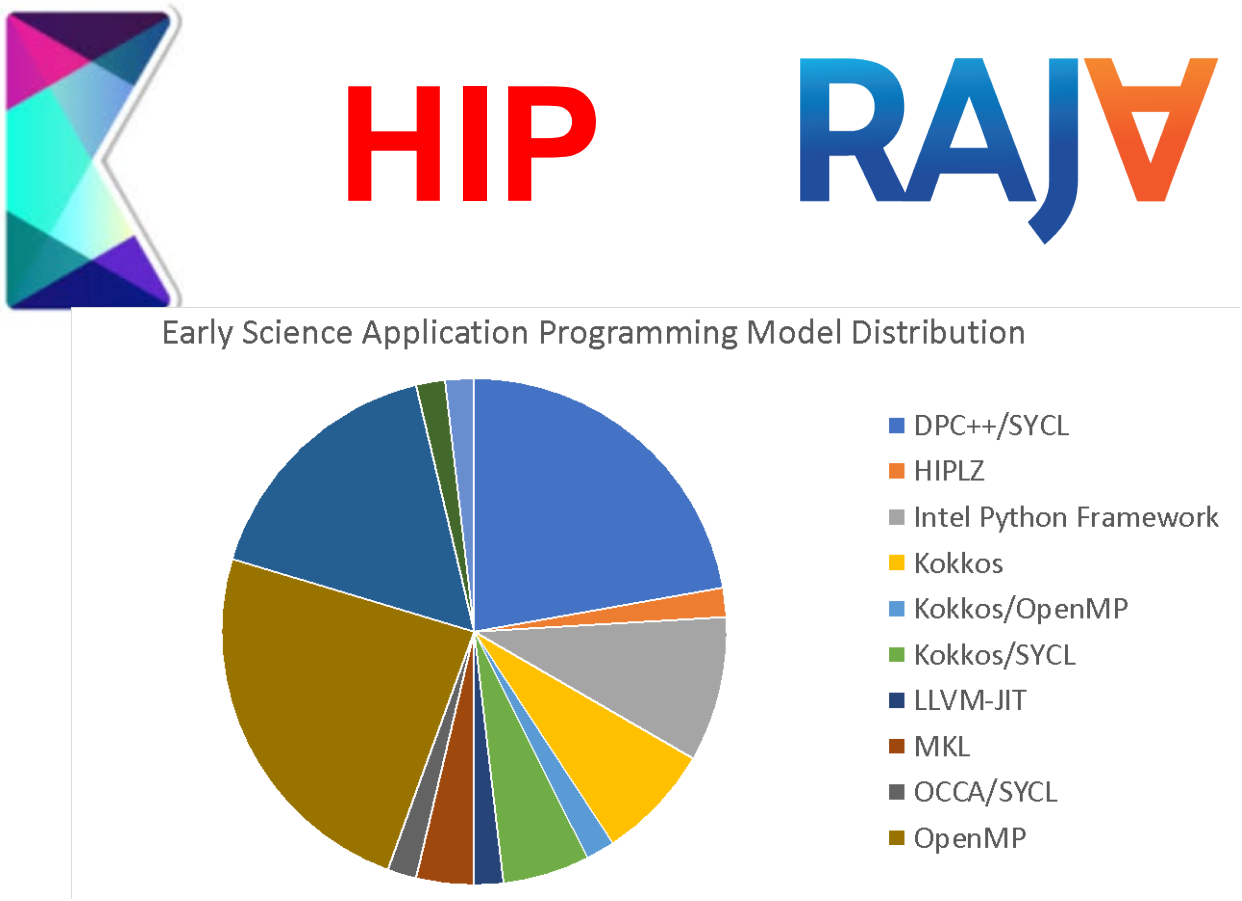
oneDAL oneMKL oneDNN

Frameworks

PyTorch TensorFlow

Available Aurora Programming Models

- Aurora applications may use:
  - DPC++/SYCL
  - OpenMP
  - Kokkos
  - Raja
  - OpenCL
- Experimental
  - HIP – *running GAMESS, CP2K, libCEED*
- Not available on Aurora:
  - CUDA
  - OpenACC



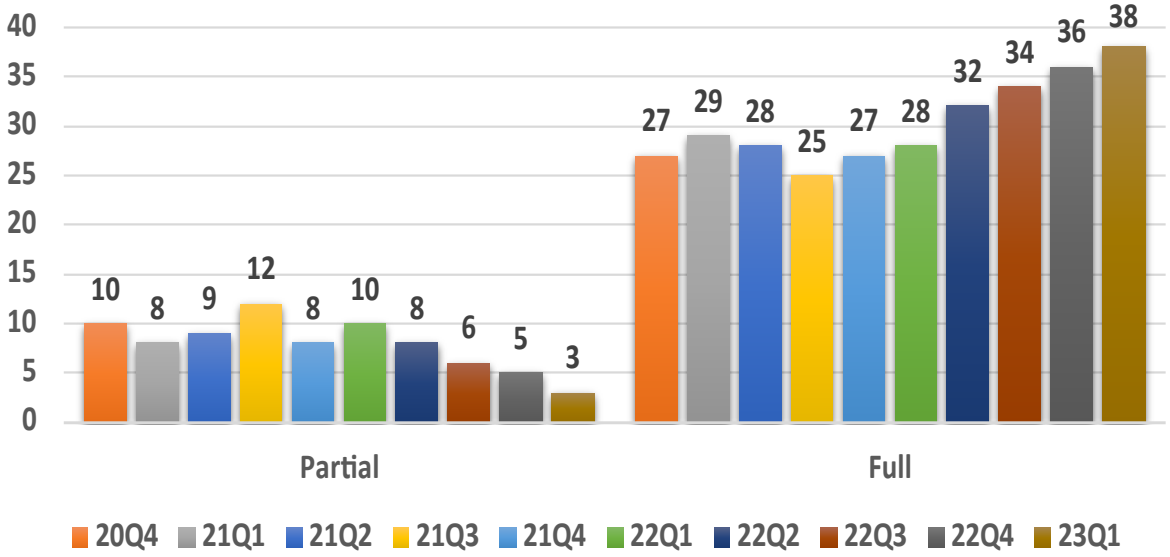
Aurora Applications Overview

- Early applications for Aurora come from two programs:
  - The Argonne Early Science Program (ESP) : 19 projects
    - 9 Simulation projects
    - 5 Learning projects
    - 5 Data projects
  - The DOE Exascale Computing Project (ECP) : 15 projects
- Some projects contain multiple codes
  - 44 codes are targeted for Aurora as part of ESP or ECP projects
    - 3 codes are intended for use on the CPU only
- Involves effort from over 60 Argonne and Intel staff and over one hundred outside collaborators
- Almost all projects involve teams from outside Argonne

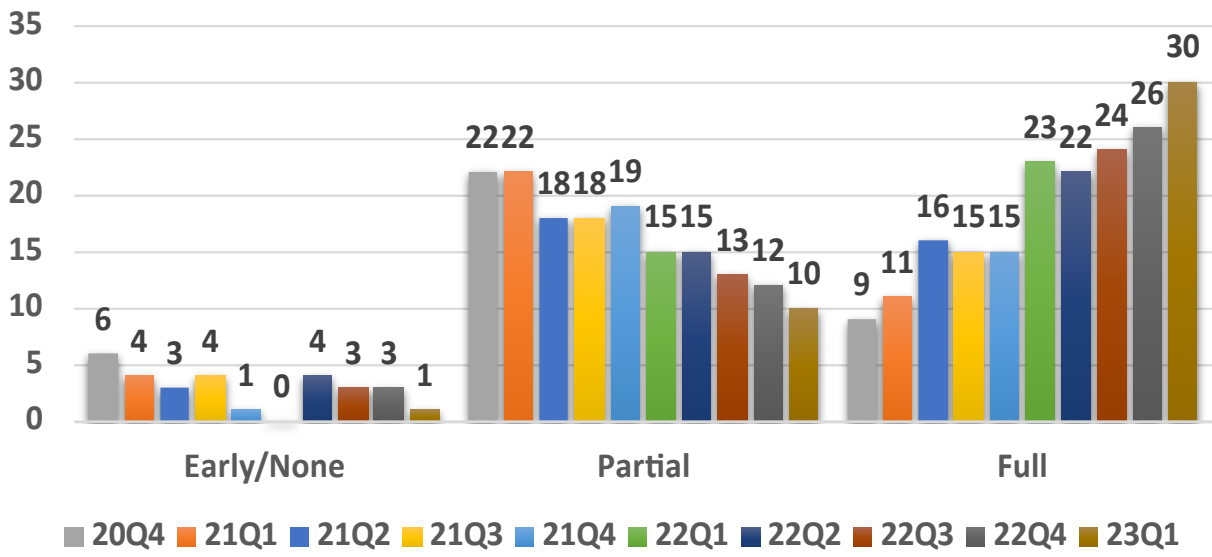
Aurora Applications Development

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

Application Science Implementation



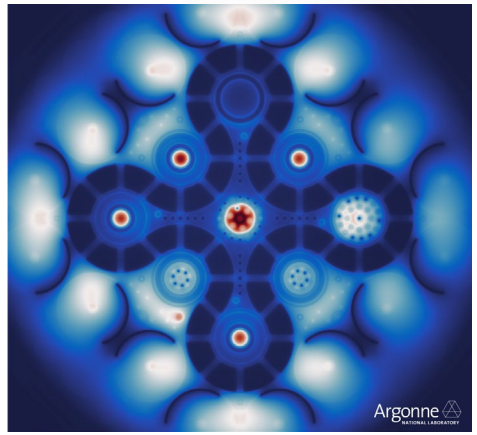
Port to Aurora Programming Models

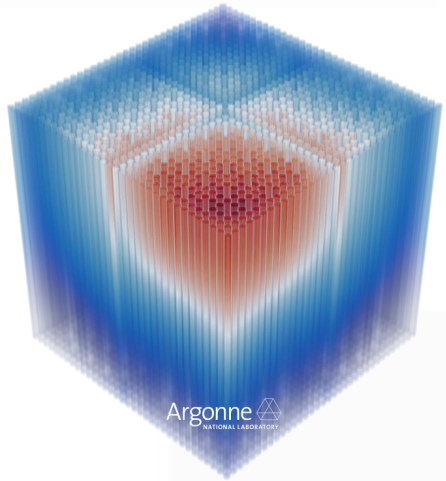


OpenMC *(courtesy of John Tramm)*

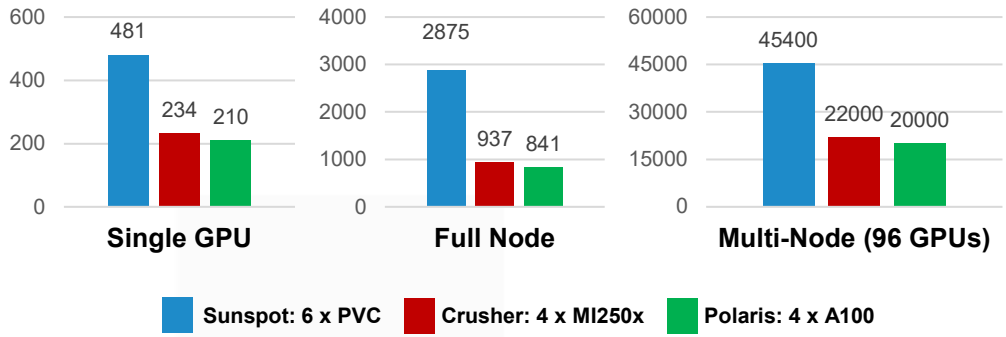
https://docs.openmc.org

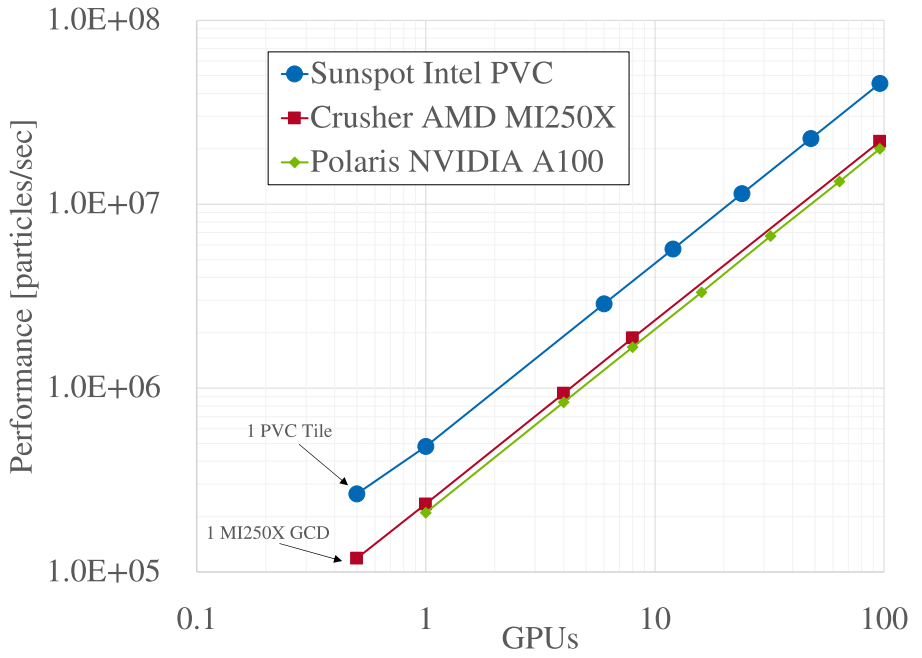
- OpenMC is being developed as part of the ECP ExaSMR project (PIs: Steven Hamilton, Paul Romano)
- OpenMC is a Monte Carlo particle transport code written in C++ and the OpenMP target offloading programming model
- The project seeks to accelerate the design of small modular nuclear reactors by generating virtual reactor simulation datasets with high-fidelity, coupled physics models for reactor phenomena that are truly predictive
- The Monte Carlo method employed by OpenMC is considered the "gold standard" for high-fidelity but these methods suffer from a very high computational cost.
- The extreme performance gains OpenMC has achieved on GPUs is finally bringing within reach a much larger class of problems that historically were deemed too expensive to simulate using Monte Carlo methods.





(Higher is better)

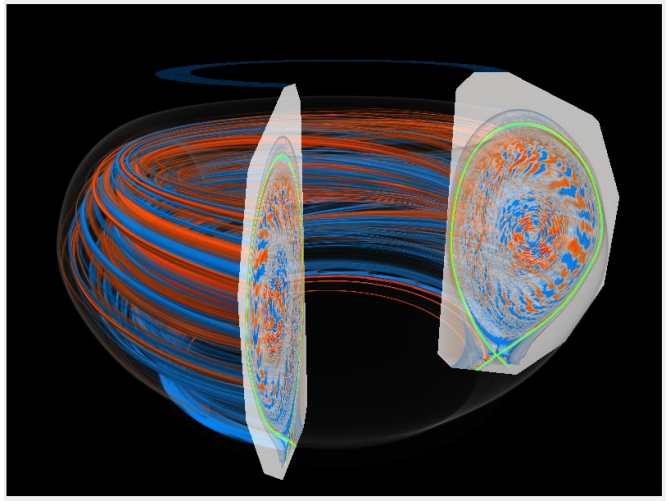


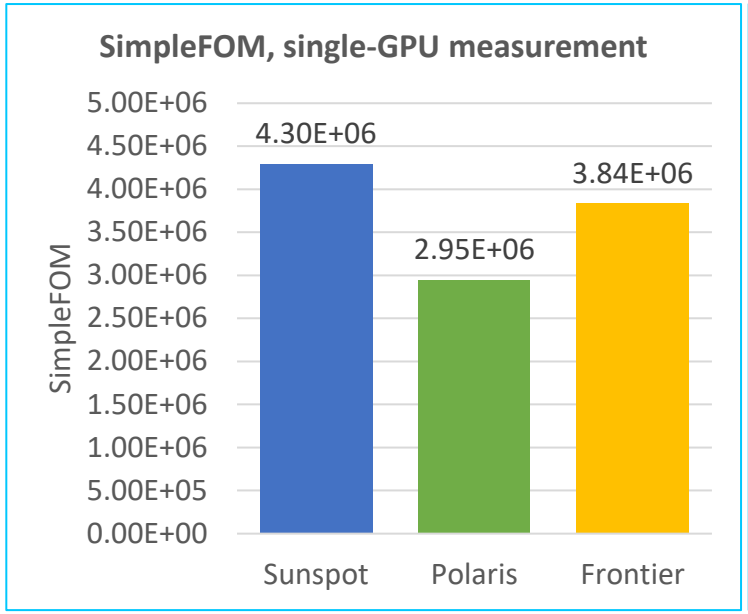


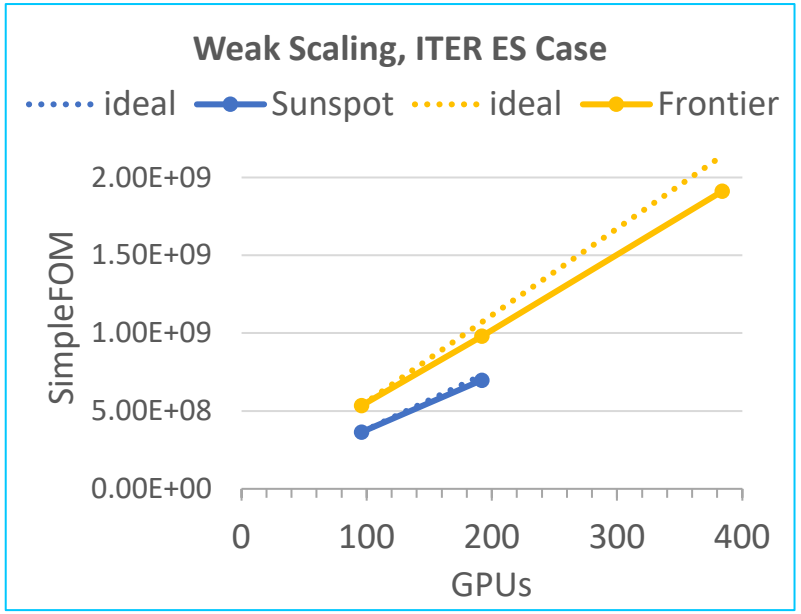
XGC *(courtesy Tim Williams, Aaron Scheinberg)*

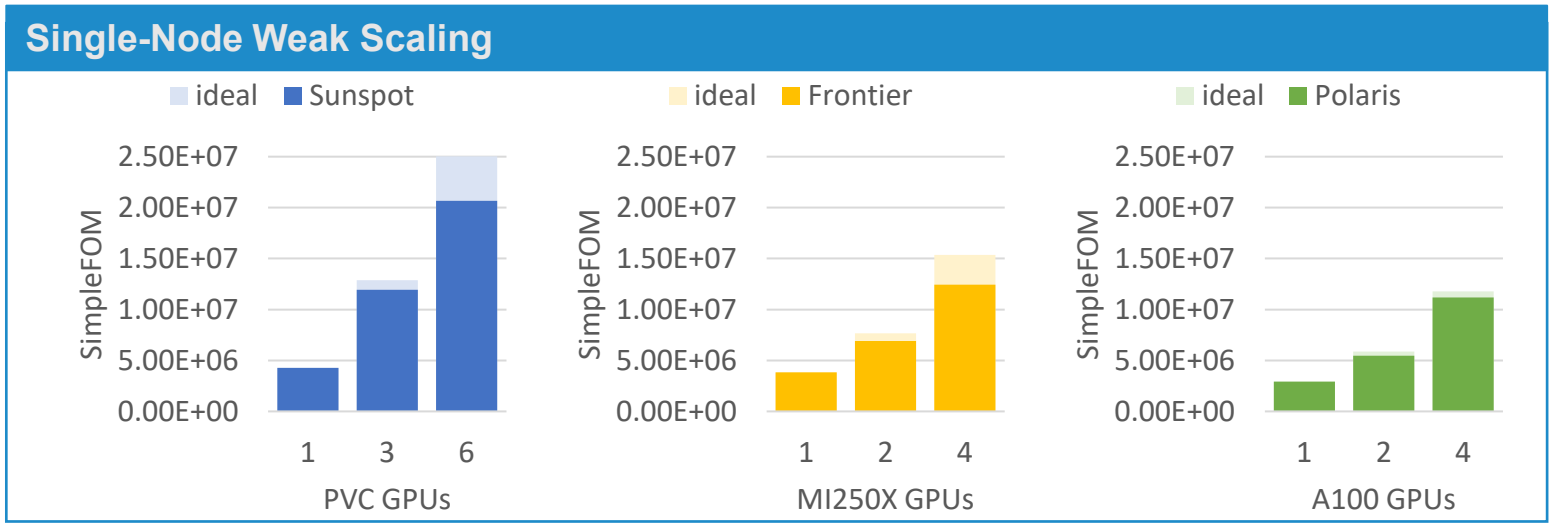
ESP Project PI: CS Chang
ECP Project PI: Amitava Bhattacharjee

- Science case: Predict ITER fusion reactor plasma behavior with Tungsten impurity ions sputtered from the divertor
- Gyrokinetic particle-in-cell simulation of tokamak plasma using C++ and:
  - Kokkos/SYCL on Intel GPUs
  - Kokkos/HIP on AMD GPUs
  - Kokkos/CUDA on NVIDIA GPUs





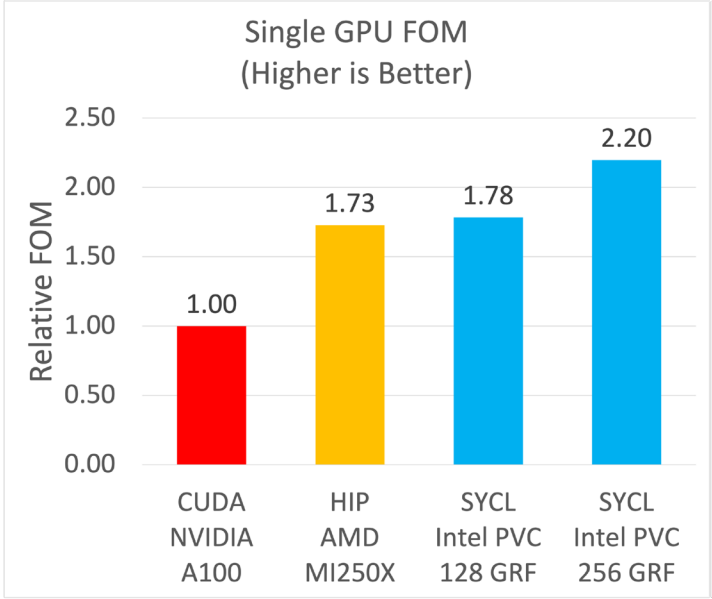


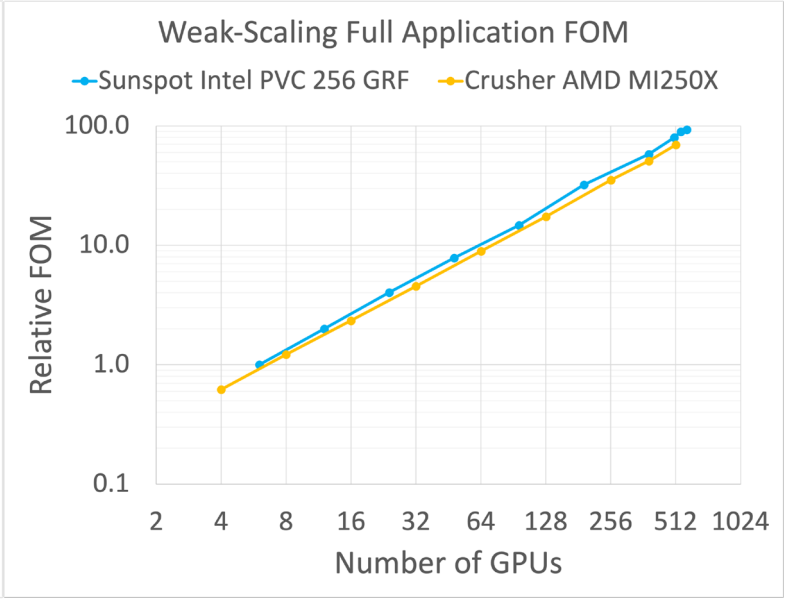


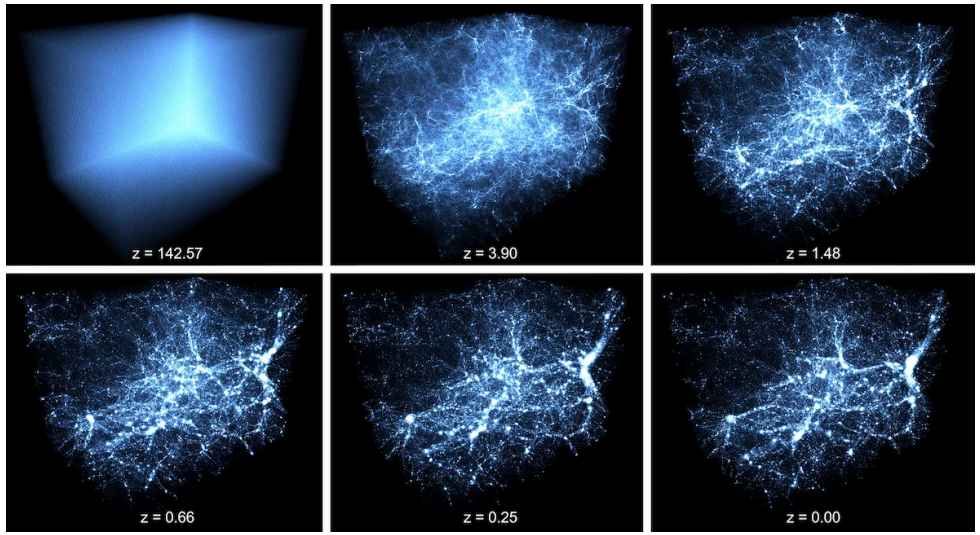
CRK-HACC *(courtesy Adrian Pope, Steve Rangel, Nick Frontiere)*

ESP/HACC PI: Katrin Heitmann
ECP/ExaSky PI: Salman Habib

- CRK-HACC simulates the formation of large-scale structures in the Universe over cosmological time.
- CRK-HACC employs n-body methods for gravity and a novel formulation of Smoothed Particle Hydrodynamics.
- CRK-HACC is a mixed-precision C++ code, with FLOPS-intense sections implemented using architecture-specific programming models in FP32 precision.







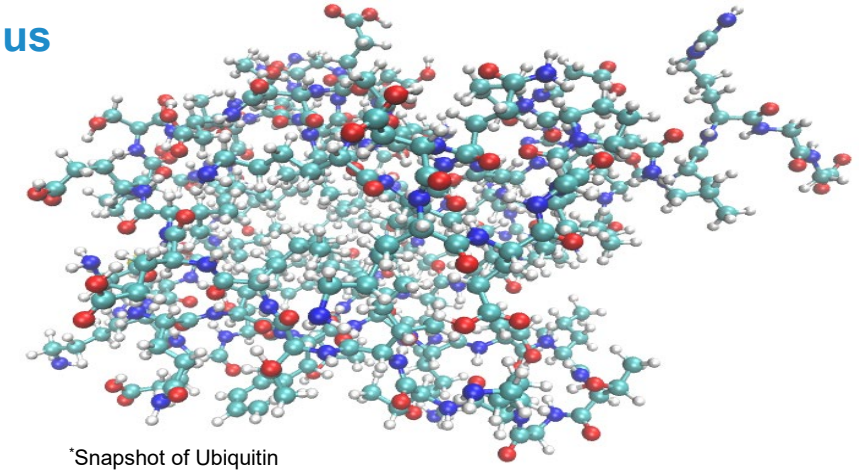
- CUDA and HIP are maintained as a single source with macros.
- SYCL kernels were translated from CUDA using SYCLomatic and custom LLVM-based tools, including optimizations for Intel GPUs.
- Figure-of-Merit (FOM) has units of particle-steps per second.
- Single GPU FOM problem used 33 million particles per GPU, and Intel PVC results are shown for both small (128) and large (256) General-purpose Register File (GRF) modes.
- Weak-scaling results are shown with the full application FOM, where the GPU represents roughly 80% of the total wall clock.

NWChemEx *(Courtesy of Ajay Panyala)*

https://github.com/NWChemEx-Project

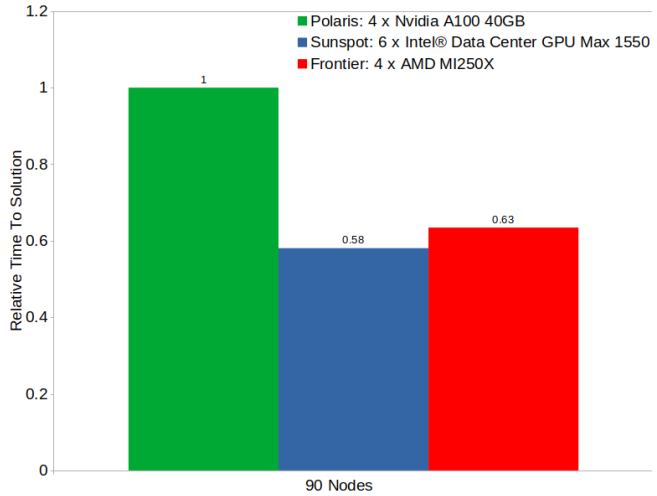
ESP & Project Project PI: Theresa Windus

- NWChemEx is a general purpose electronic structure code, which includes
  - Array of high-fidelity coupled cluster methods
  - Hartree-Fock, DFT, MP2 methods
  - Reduced-scaling DLPNO formulation
  - Molecular dynamics
- Programming models: C++, CUDA, HIP, SYCL
  - Communication frameworks: Global Arrays, UPC++, MADNESS
  - Tensor Contraction Engines: TAMM, TiledArray
- Key physics modules
  - DLPNO-CCSD(T)
    - Reduced-scaling implementation for GPU platforms

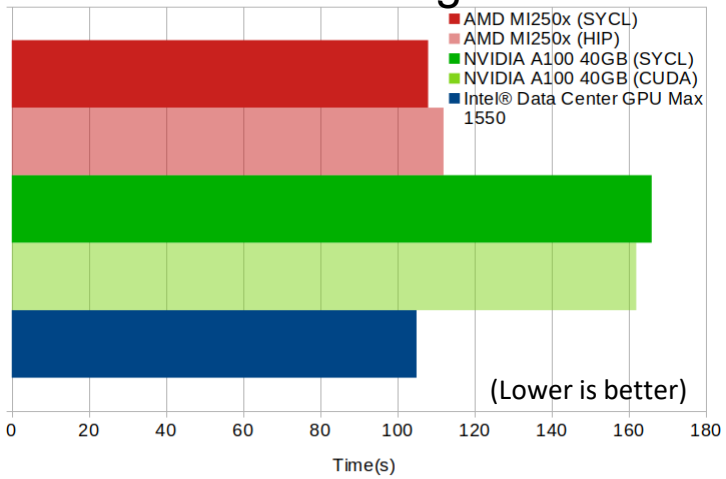


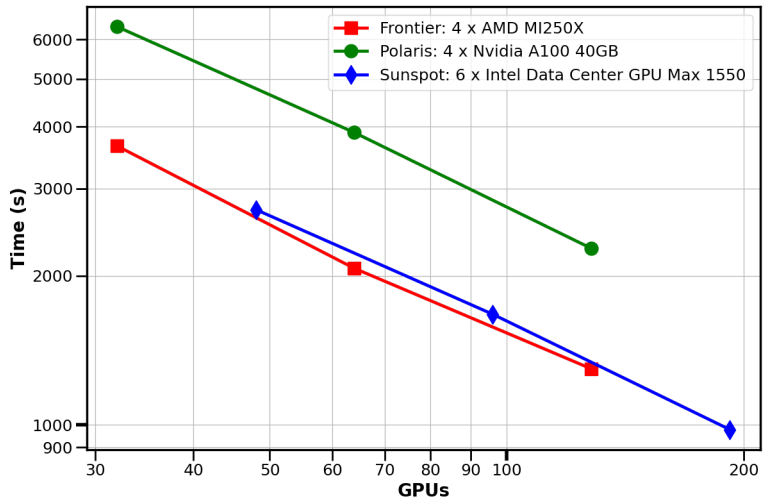
*Snapshot of Ubiquitin Protein

Strong Scaling Performance on 90-nodes



Performance on Single GPU





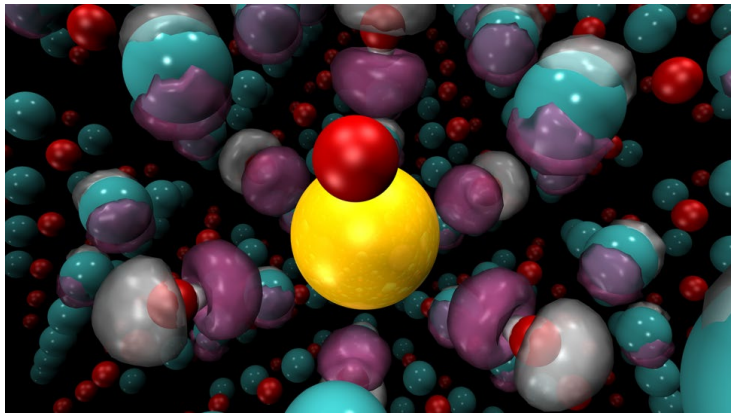
- Single GPU, Time in seconds for DLPNO-CCSD per iteration
- Performance of SYCL on NVIDIA & AMD were comparable with native CUDA & HIP respectively

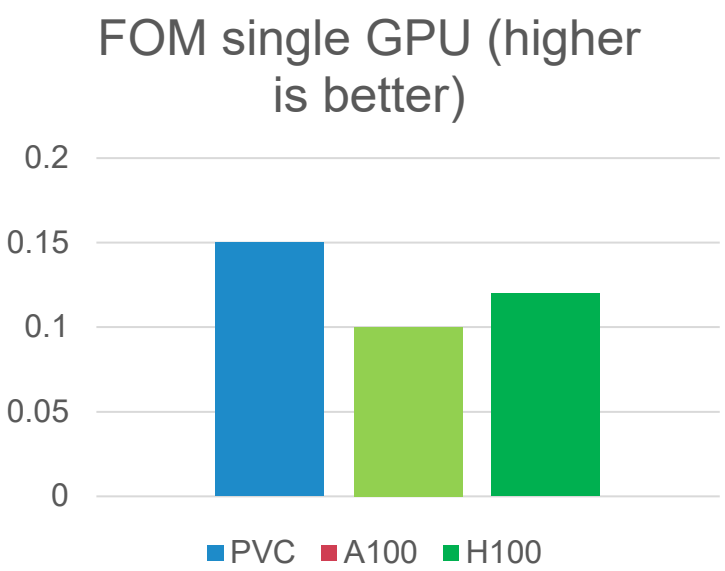
- Acknowledgment: Work performed by the NWChemEx team members without any architecture specific optimizations

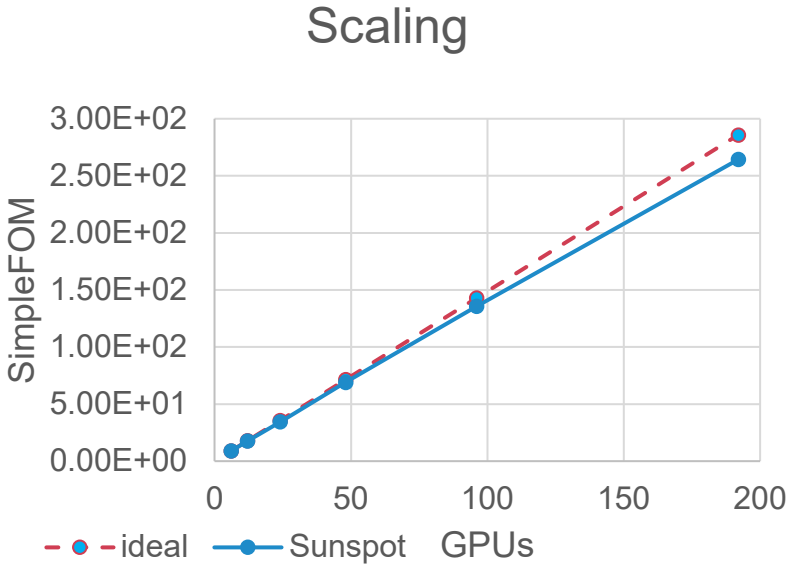
QMCPACK *(courtesy Thomas Applencourt, Ye Luo, Jeongnim Kim)*

ECP Project PI: Paul Kent

- QMCPACK, is a high-performance open-source Quantum Monte Carlo (QMC) simulation code.
- Science case: 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 (eg SYCL) around vendor optimized linear algebra.







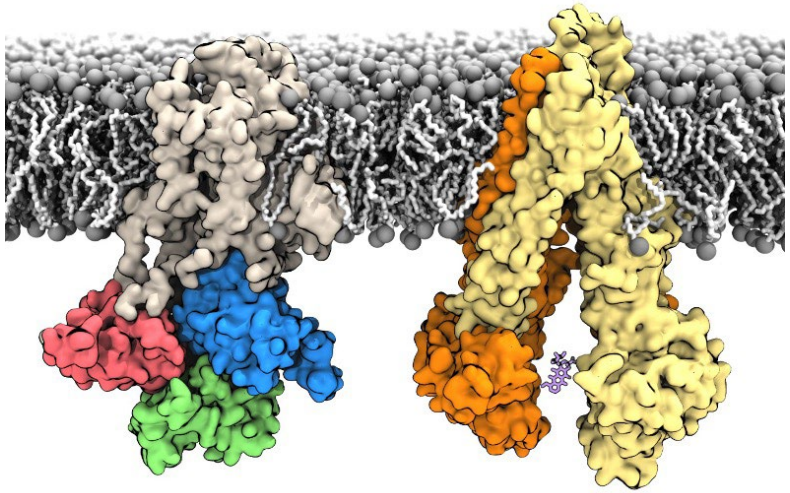
- Running `dmc-a512-e6144-DU64` problem. This simulates a supercell of nickel oxide with 6144 electrons and 512 NiO atoms total.
- Intel® Data Center GPU Max Series: 2 MPI ranks per GPU, 8 Walkers per rank, 64 GB of HBM per stack. Using Intel(R) oneAPI DPC++/C++ Compiler 2022.12.30
- A100 (40GB): 1 MPI Rank, 7 Walkers. LLVM15 compiler.
  H100: llvm/clang 17, cuda 11.8): 1 MPI Rank, 7 Walkers
- The Figure Of Merit (FOM) measure is throughput (walker moves/second). Higher is better.

NAMD 2 *(Courtesy of Wei Jiang)*

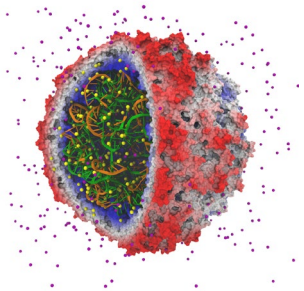
Scalable molecular dynamics for exascale computations

ESP PI: Benoit Roux

- Simulate large biomolecular systems or complex macromolecular machines
- Science problem: molecular structure-function relationship
- Algorithm: particle motion integration with short- and long-range force calculation
- Fine-grained force-domain decomposition
- Written using C/C++, Charm++, CUDA, HIP, SYCL



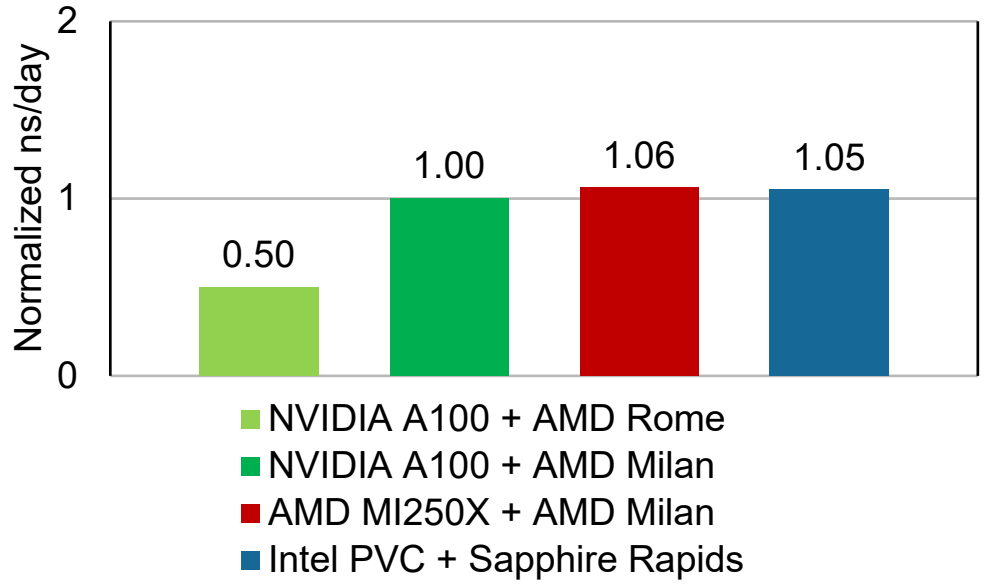
NAMD website: https://www.ks.uiuc.edu/Research/namd



Benchmarking NAMD 2.15alpha2 on STMV (1.06M atoms) NVE simulation: *CHARMM force field (12A cutoff), rigid bonds with 2 fs timestep, multiple time stepping with 4 fs PME*

(Higher is better)

Single-GPU Results



NAMD GPU-offload performance depends on both GPU and CPU performance together with host-device latency and bandwidth

Questions?

This publication used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.