IXPUG Workshop HPC Asia 2023

Porting Simulation, Data-Intensive, and AI Applications to the Aurora Exascale System

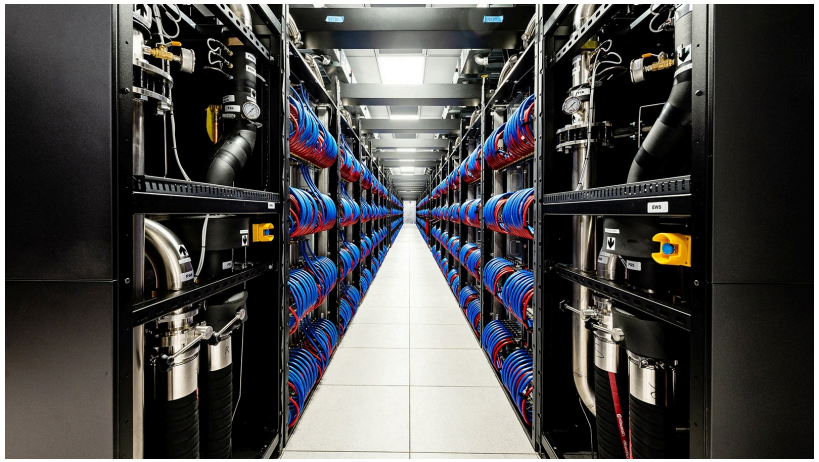
TIM WILLIAMS
Deputy Division Director
Computational Science Division (CPS)
Argonne National Laboratory

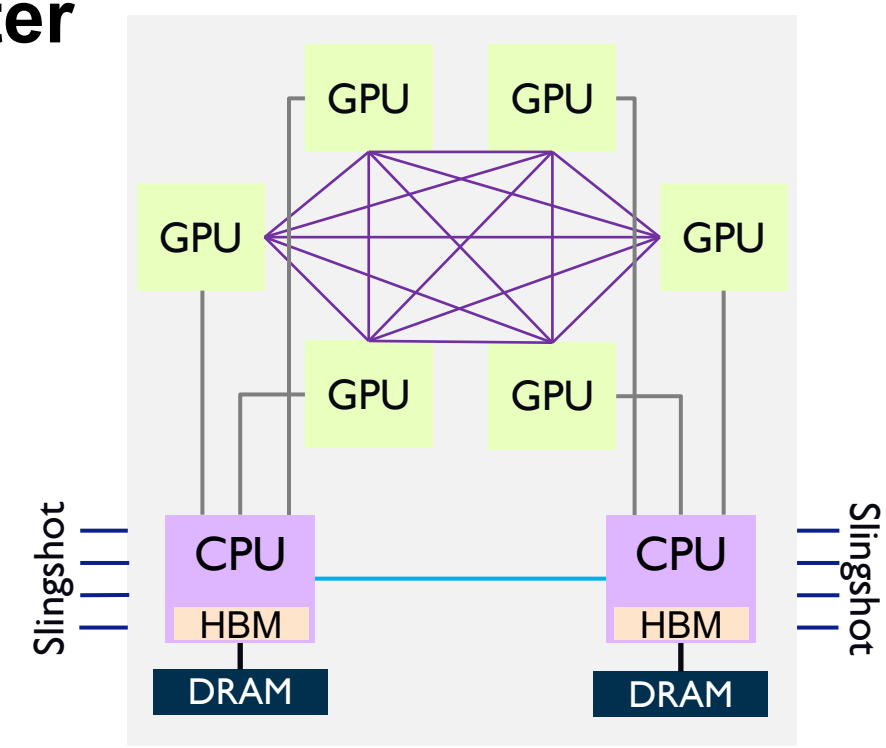
VENKAT VISHWANATH
Data Science Team Lead
Argonne Leadership Computing Facility (ALCF)
Argonne National Laboratory

SCOTT PARKER
Performance Engineering Team Lead
Argonne Leadership Computing Facility (ALCF)
Argonne National Laboratory

27 February 2023

Aurora – Argonne's Exascale Supercomputer





- 2 exaFLOPS double precision
- > 10,000 nodes
- HPE Slingshot 11 network, dragonfly
- > 10 PB memory

Aurora node

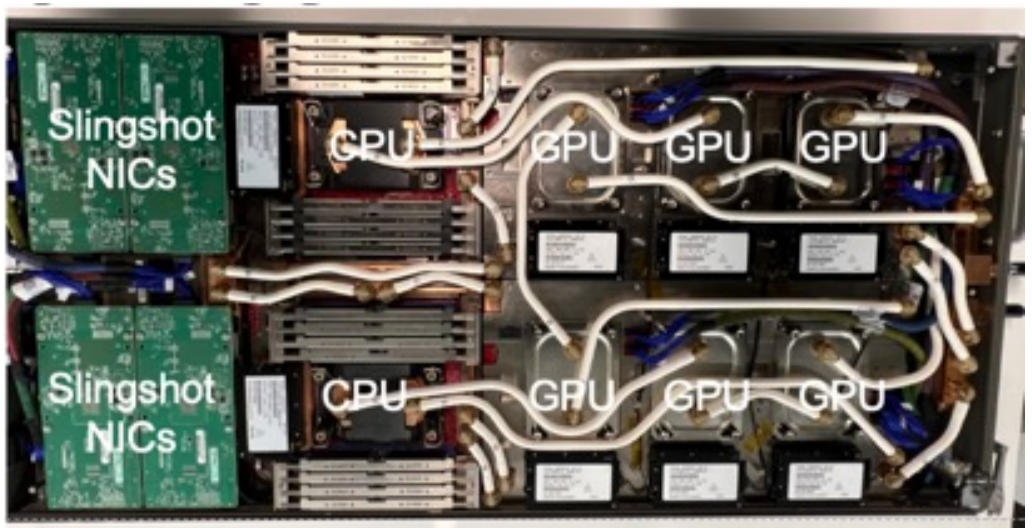
- 2X Intel® Xeon® CPU Max Series
- 6X Intel® Data Center GPU Max Series

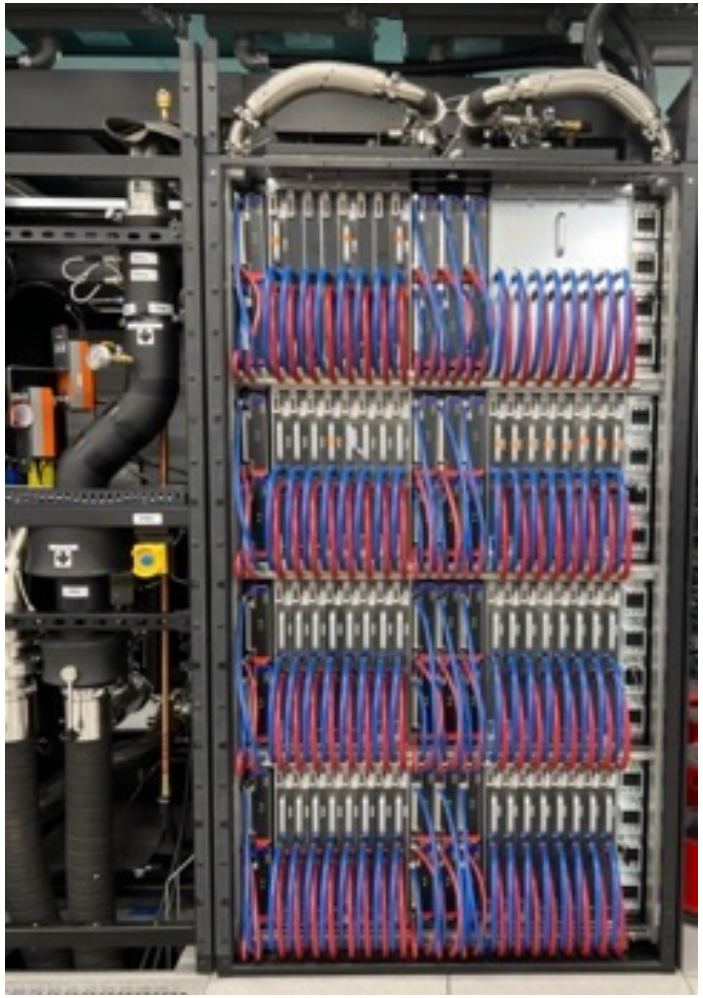
Aurora I/O System

- Distributed Asynchronous Object Storage (DAOS)
- ≥ 230 PB, ≥ 25 TB/s

Status of Aurora

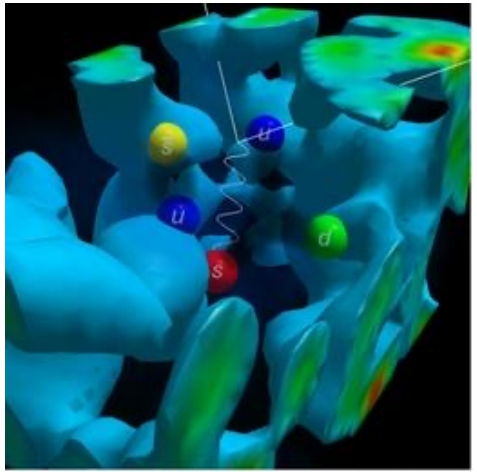
- Aurora is built with exception of compute blades
- Installation of compute blades well underway
- Targeting early-user system access in Q3

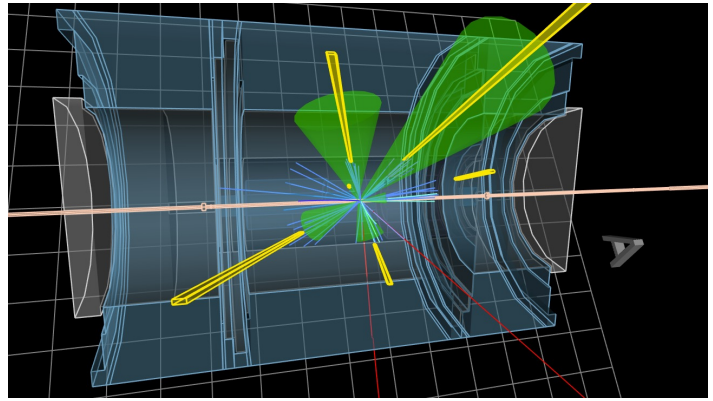


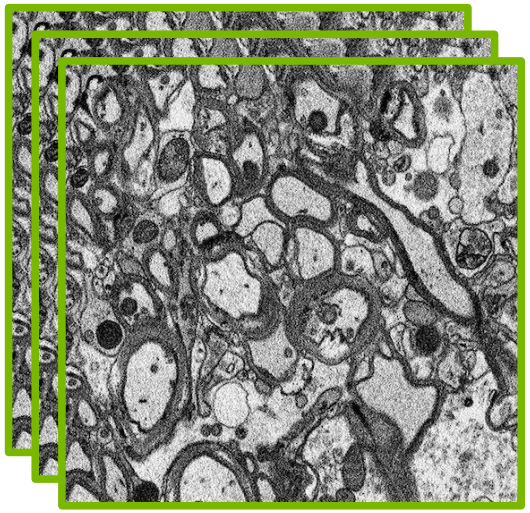


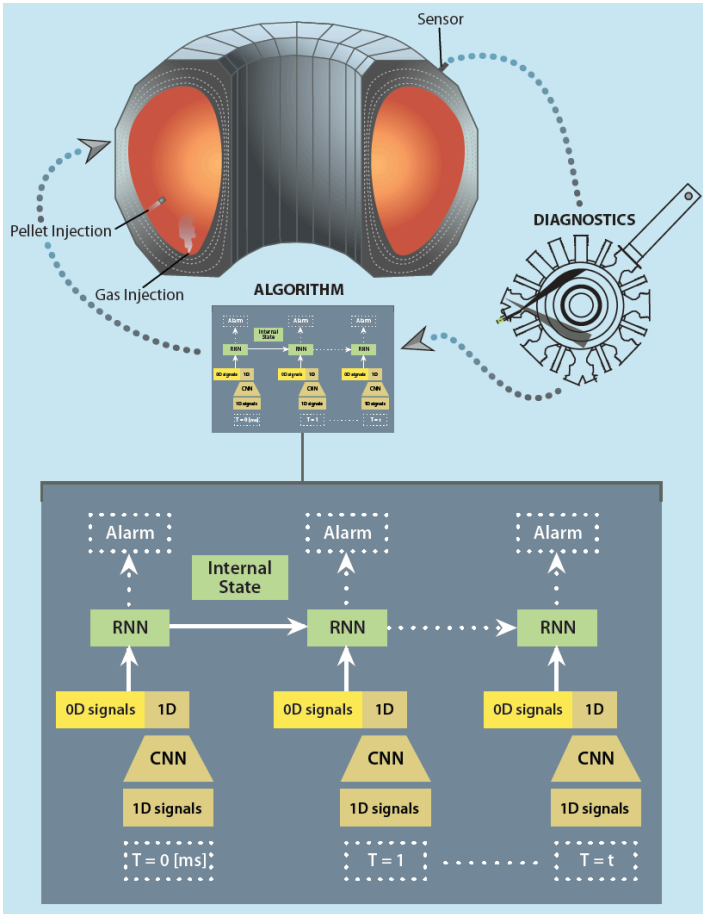
Exascale Science

Simulation | **Data** | **Learning**







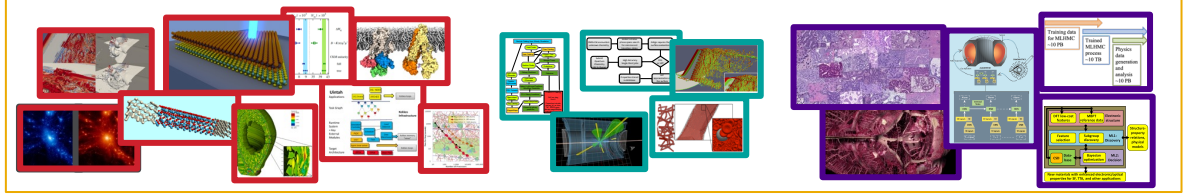


- Exascale CFD simulation data trains NN model for rapid design studies
- Exascale cosmology simulation data, sky survey data, and ML inform observations and test theory
- Exascale training from tokamak experimental data for inference at experiment
- Exascale inference from DNN to construct connectome from massive EM brain tissue dataset
- Exascale search spaces: PV materials, chemical kinetics, cancer drug combinations, detector collision events

Exascale Applications/Software Readiness

AURORA | EARLY SCIENCE PROGRAM

- ALCF Aurora Early Science Program (ESP)
- *Managers: Tim Williams, Venkat Vishwanath*
- 9 Simulation,10 Data and Learning projects
- **Every project will run a proposed science campaign on Aurora**
- Training: Workshops, Hackathons, Dungeon Sessions, webinars
- Argonne postdoc and staff support (Catalysts)



ECP EXASCALE COMPUTING PROJECT

- DOE Exascale Computing Project (ECP)
- 3 technical areas: Application Development, Software Technology, Hardware and Integration
  - AD: 21 applications projects preparing codes for exascale
  - ST: 66 unique software products
  - HI: Applications Integration: deploy apps on specific exascale systems (Aurora, Frontier)
- AppInt funding for Argonne staff for Aurora:
  - ALCF working with 15 ECP AD so far

Argonne-Intel Center of Excellence – dedicated Intel staff

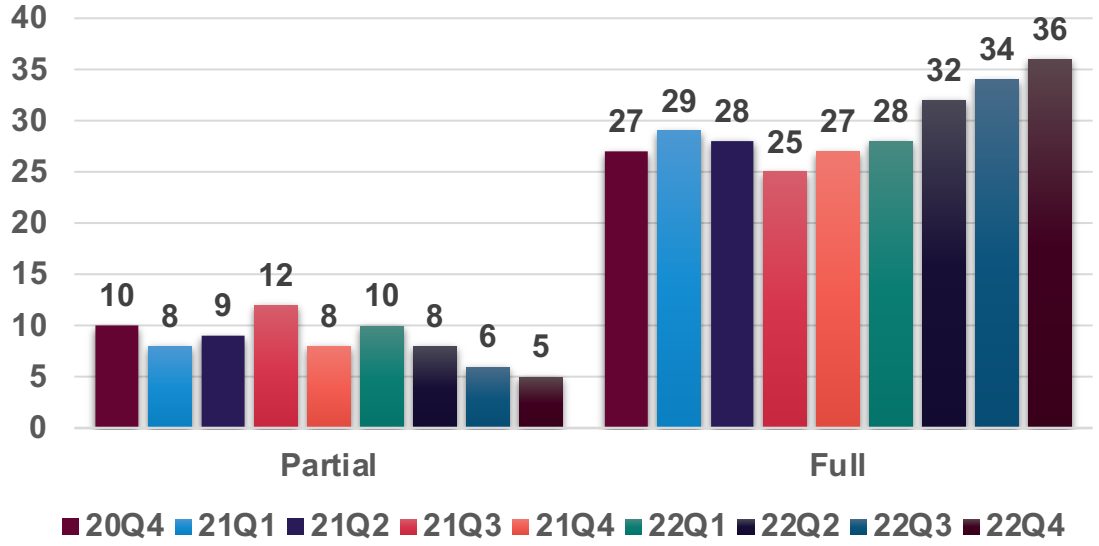
http://esp.alcf.anl.gov

Tracking Aurora Applications Development

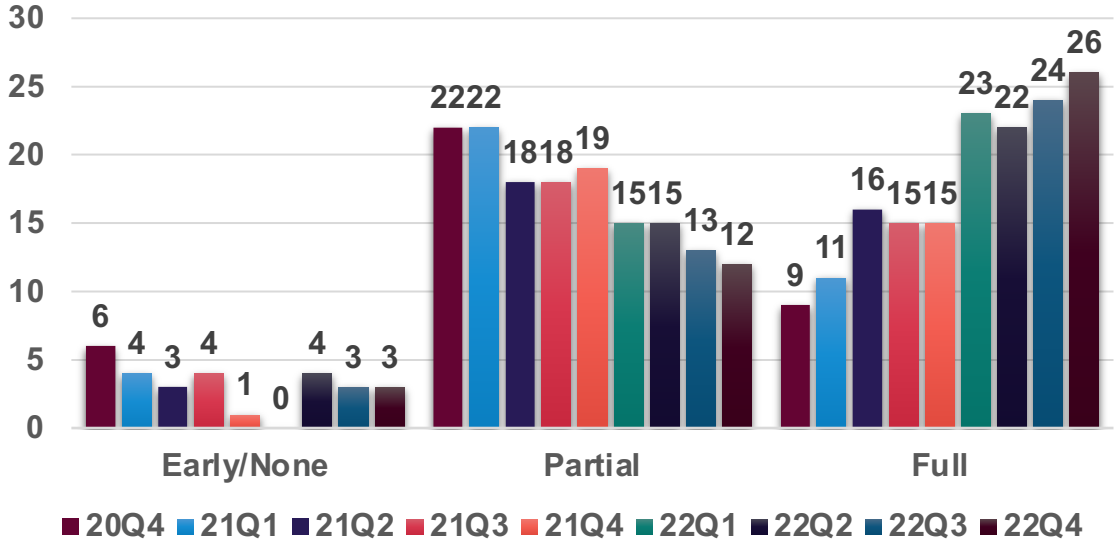
- Steps in application preparation
  - Implementation of science and algorithms
  - Porting to Aurora programming models
  - Testing with Aurora SDK on Aurora testbeds
  - Tuning for performance on Aurora testbeds
  - Scaling across the Aurora system

- ALCF and Intel working with over 40 projects to ready codes for Aurora
- Effort from over 60 Argonne & Intel people and numerous outside teams

Application Science Implementation



Port to Aurora Programming Models



2022 Q4 Aurora Applications Status

Application	Q4
HACC	
OpenMC	
XGC	
QMCPack	
AMRWind	
NAMD	
QUDA	
LAMMPS	
NekRS	
SW4	
FloodFillNetwork	
Data Driven CFD	
Harvey	
PHASTA	
MFIX-Exa	
FushionDL	
DCMesh	
E3SM-MMF	
CANDLE/UNO	
Thornado	
Chroma	

Application	Q4
GENE	
BerkelyGW	
Latte	
cctbx	
MILC	
NWChemEx	
MadGraph	
Grid	
GAMESS	
NYX	
DarkSkyMining	
Uintah	
Nalu-Wind	
GEM	
mb_aligner	
RXMD-NN	
Flow Based Generative Model	
FastCaloSim	
spiniFEL	
Multi-Grid Parameter Opt.	

Running
Running
Running
Partially Running
Porting in Progress

Exascale Programming

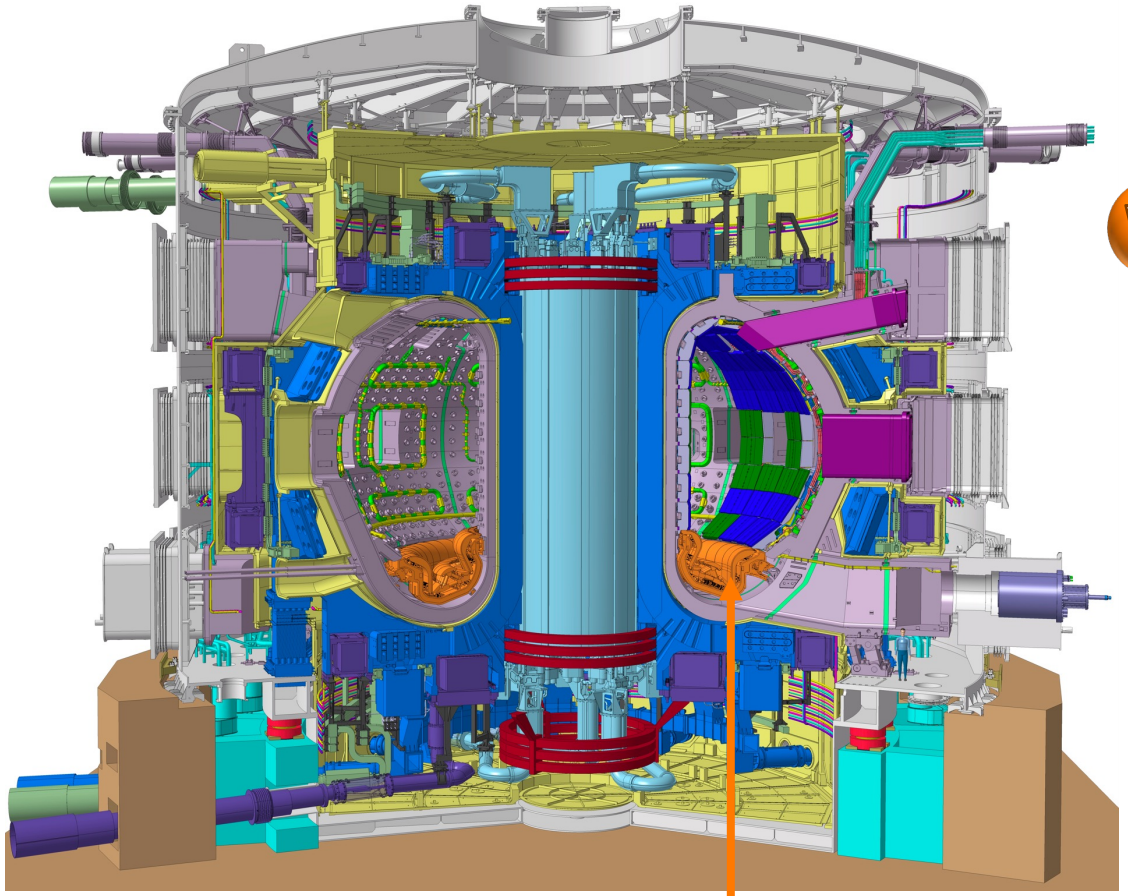
- We've been developing for future target exascale architectures for years
  - GPU acceleration
- Performance portability across Intel, AMD, NVIDIA
  - Pick a portability layer (Kokkos, SYCL, OpenMP 5, your own library, TensorFlow, PyTorch)
  - Work with implementers of layer on target systems
- Lingua Franca
  - Simulation – compiled languages; math libraries; SYCL, CUDA, HIP, (Kokkos, Raja)
  - Data & Learning – Python frameworks

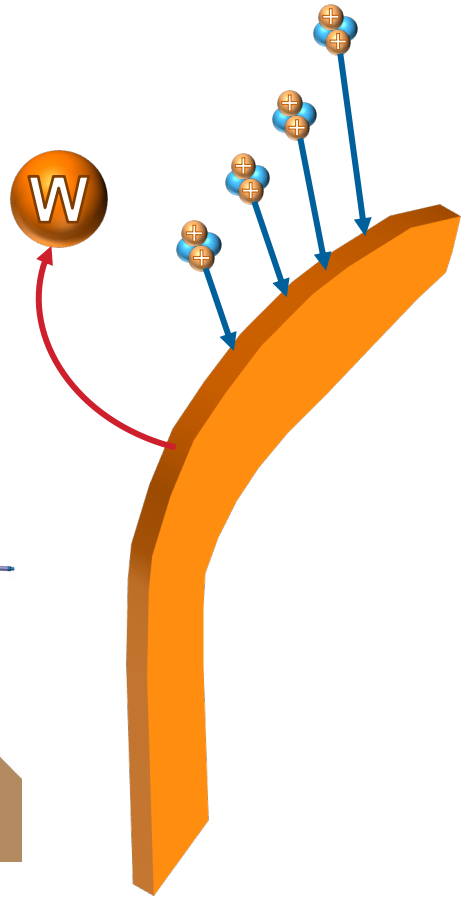
- Aurora Programming
  - Simulation
    - oneAPI, oneMKL
    - ECP E4S
  - Data & Learning
    - Frameworks backed by oneDAL, oneCCL
    - DAOS

- Aurora Development Hardware
  - In the beginning, Xeon integrated graphics
  - Sequence of pre-production Intel discrete GPUs
  - Sunspot
    - 128 nodes of Aurora hardware

XGC Early Science Project *(Simulation; PI: CS Chang)*

ITER Tokamak

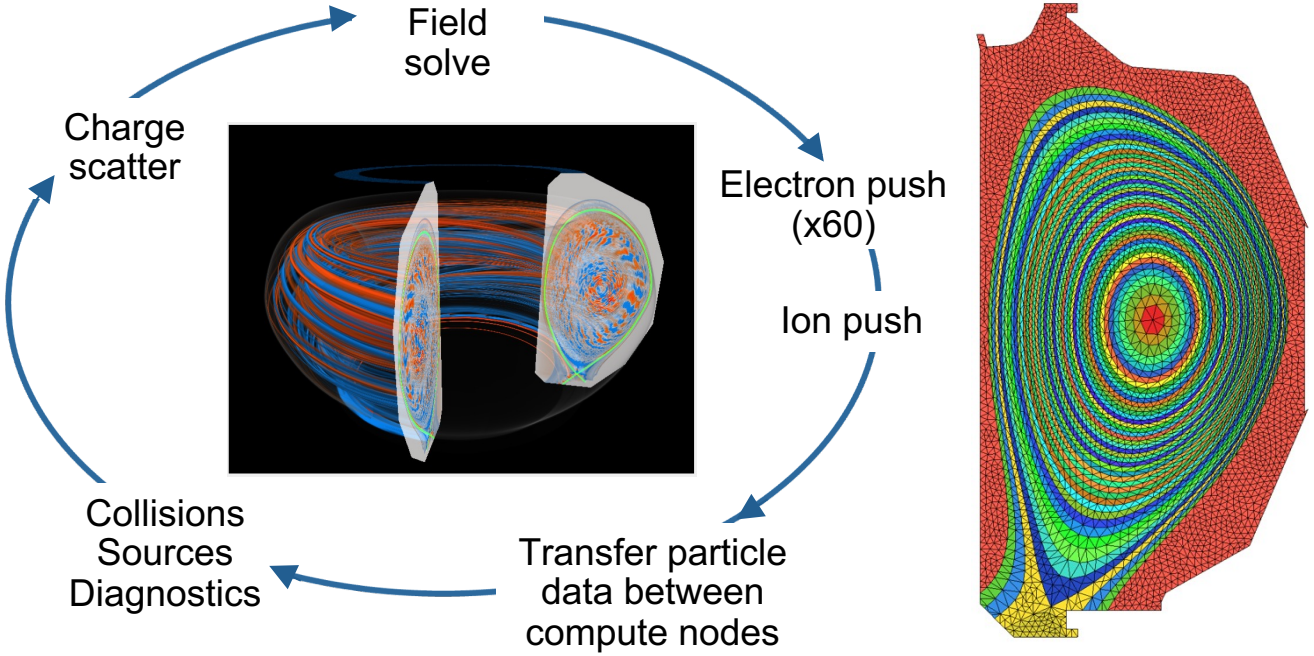




- Hot exhaust such as He particles hits the divertor
- Knocks out Tungsten atoms (sputtering)
- Some are ionized and re-enter the fusion plasma
- Interacts, impacts turbulence and other plasma behaviors
- Will it negatively impact confinement or energy production?
- Early Science campaign: Predict ITER plasma behavior with Tungsten impurity ions

XGC Early Science Project

- XGC gyrokinetic PIC (Particle-in-Cell) fusion plasma physics simulation on unstructured grid
  - Multi-ion-species
- Aurora implementation: Kokkos library with SYCL execution space
  - Portable across {Intel, AMD, NVIDIA} GPUs



Single GPU Performance FOM

- SimpleFOM = 1/(loop_time/(#particles × #timesteps))
- 2 MPI ranks
- XGC1 small test case
- 80 million particles

GPU	SimpleFOM
NVIDIA A100 (Polaris)	1.4×10^6
Intel® Data Center GPU Max Series (Sunspot)	2.2×10^6

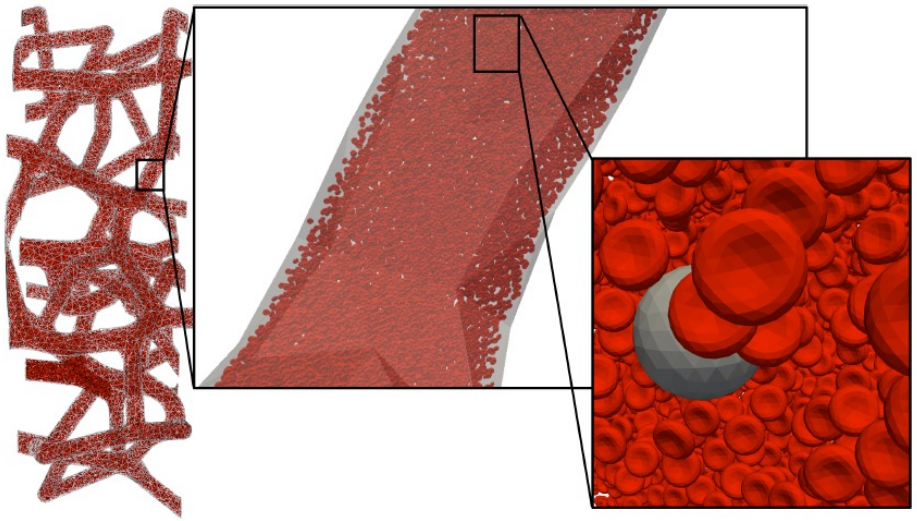
Argonne POCs: T. Williams, A. Scheinberg
Intel POC: R. Bustamante

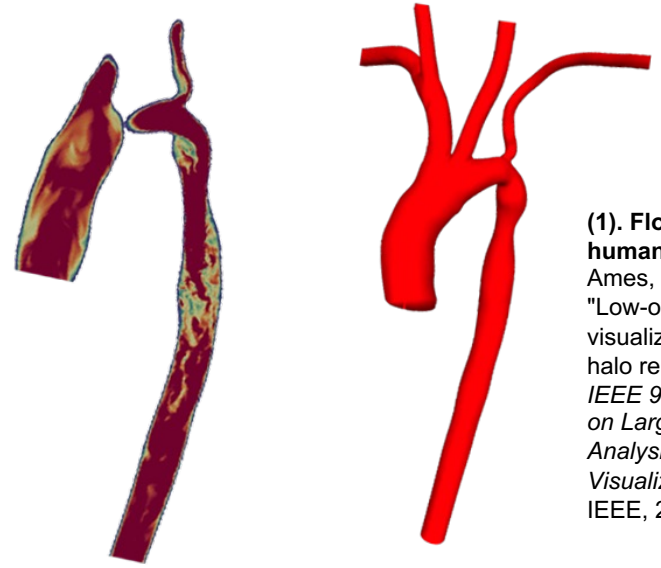
Cancer Metastasis Early Science Project *(Data: PI: Amanda Randles)*

- **ESP GOAL**: Investigate circulating tumor cells in human vasculatory system
  - 3D Lattice Boltzmann CFD
  - FEM for cells, deformation
  - Immersed boundary method to couple
- In situ exascale data visualization & analysis
- Aurora implementation: bake-off between
  - SYCL (oneAPI DPC++)
  - Kokkos with SYCL backend
- HARVEY application translated with Syclomatic, runs on Intel discrete GPUs
- MiniApp LBM shows Kokkos-SYCL competitive with handcoded SYCL

Argonne POCs: J. Insley, S. Rizzi, V. Mateevitsi, G. Liu, S. Patel

Intel POC: V. Madanath





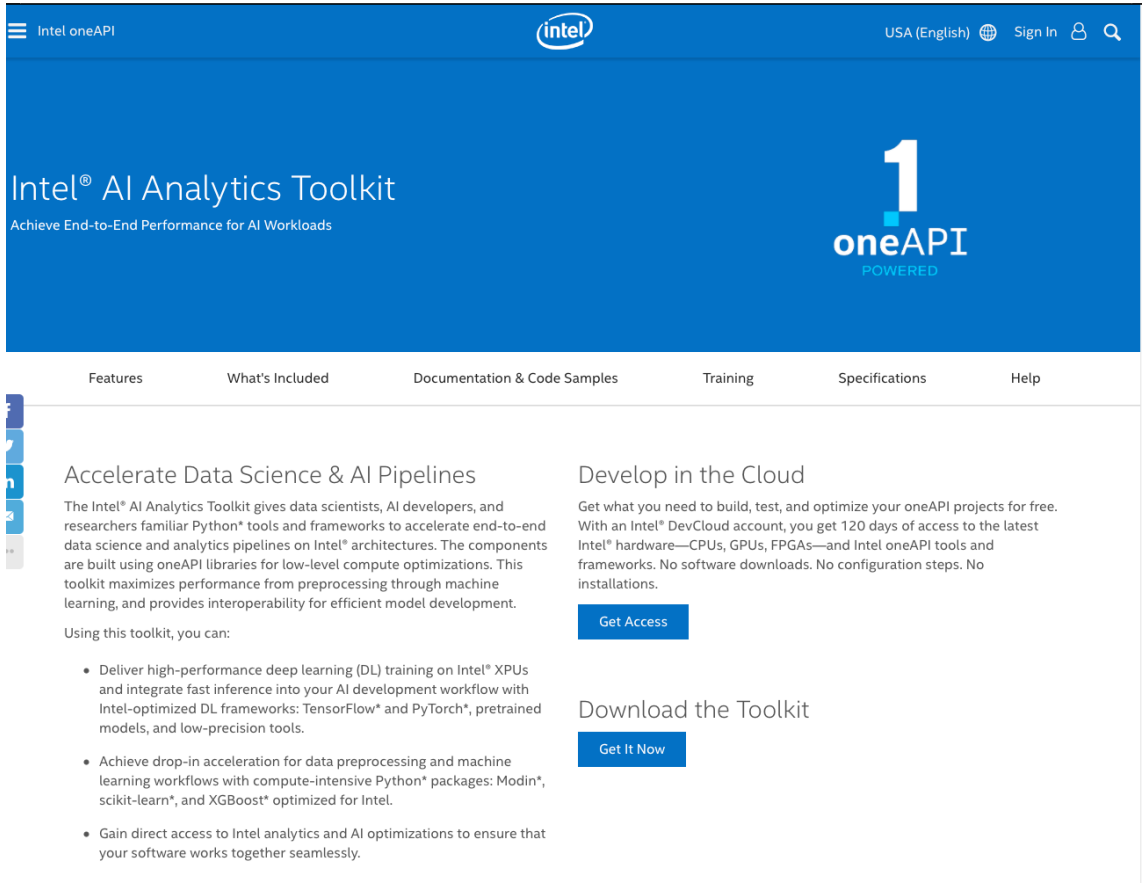
(1). Flow inside a human aorta. Ames, Jeff, et al. "Low-overhead in situ visualization using halo replay." *2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE, 2019.

Data Science and Learning on Aurora

Aurora will provide for a familiar, productive and performant HPC and AI software stack

- Python and Productivity Languages
  - Numba, NumPy, etc.
  - JAX and Julia
- Deep Learning Frameworks:
  - PyTorch, TensorFlow, Horovod, DDP, Deepspeed
- Machine Learning
  - OneDAL, scikit-learn, XGBoost, etc.
- Optimized and scalable communication using OneCCL
- Spark BigData Analytics stack
- Profiling and debugging tools

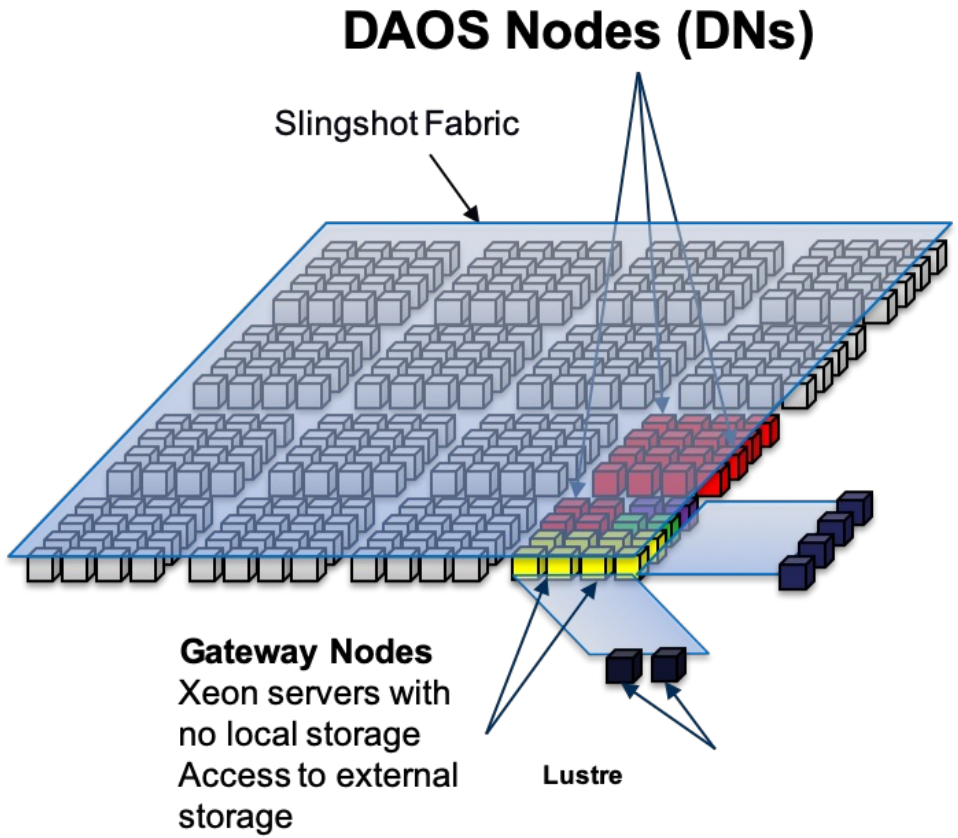
Intel AI Analytics Toolkit



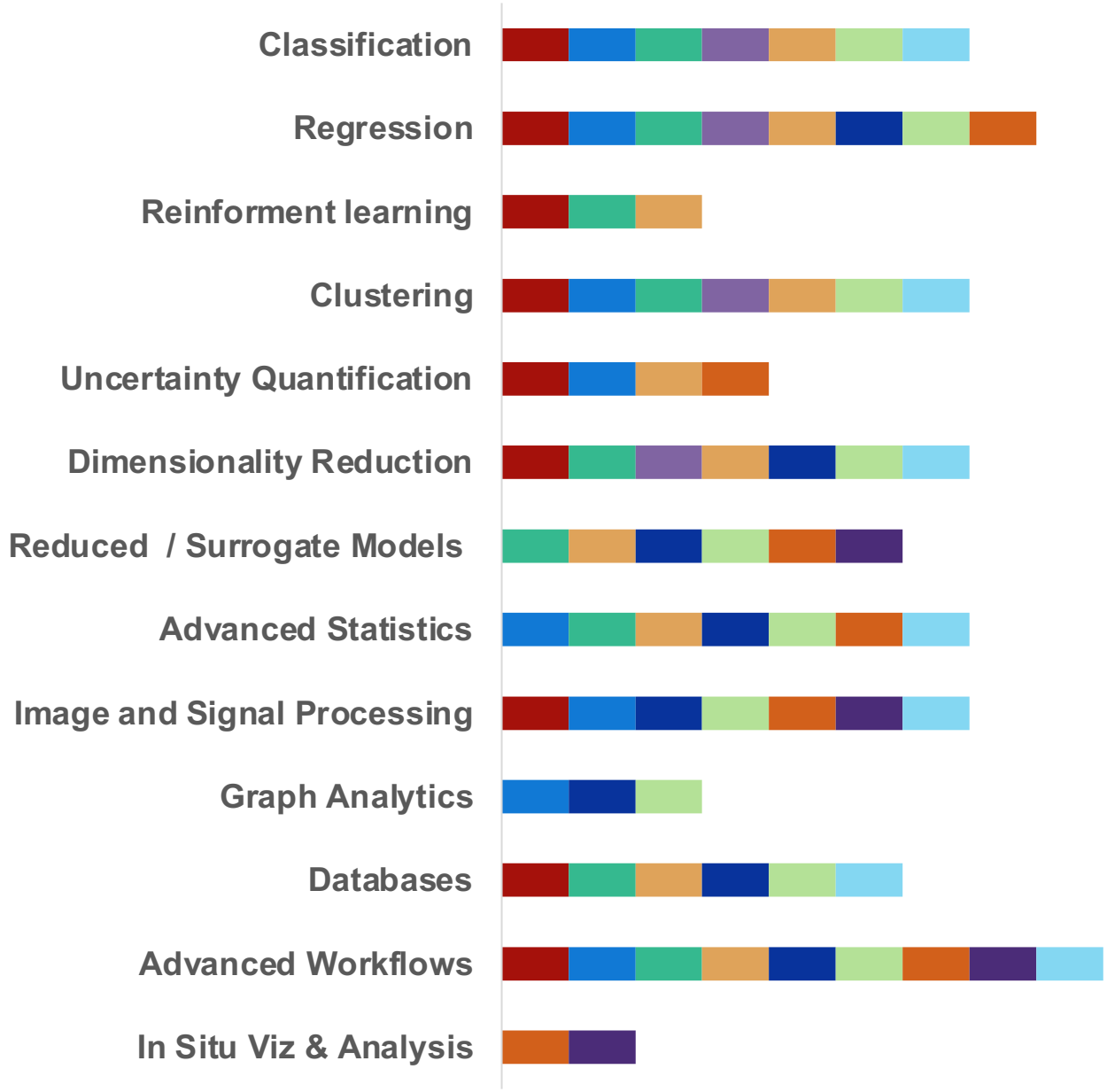
https://software.intel.com/content/www/us/en/develop/tools/oneapi/ai-analytics-toolkit.html

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



AURORA ESP Data and Learning Projects and Methods



- Virtual Drug Response Prediction
- Enabling Connectomics at Exascale to Facilitate Discoveries in Neuroscience
- Machine Learning for Lattice Quantum Chromodynamics
- Accelerated Deep Learning Discovery in Fusion Energy Science
- Many-Body Perturbation Theory Meets Machine Learning
- Exascale Computational Catalysis
- Dark Sky Mining
- Data Analytics and Machine Learning for Exascale CFD
- In Situ Visualization and Analysis of Fluid-Structure-Interaction Simulations
- Simulating and Learning in the ATLAS detector at the Exascale

Learning

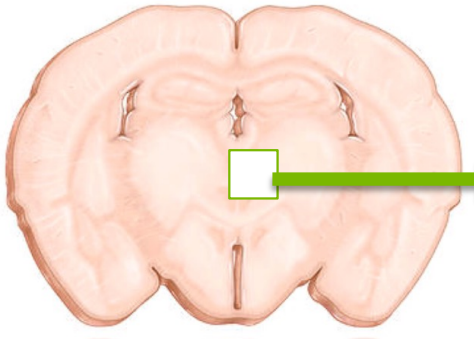
Data

Data from Gregg Wildenberg, Kasthuri Lab, UChicago

CONNECTOMICS DATA-DRIVEN MODELS

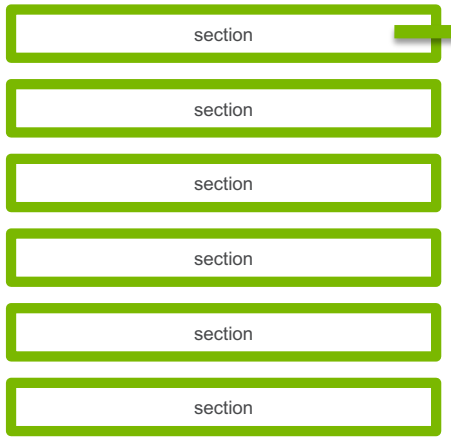
Mouse

Mouse brain: 70M neurons

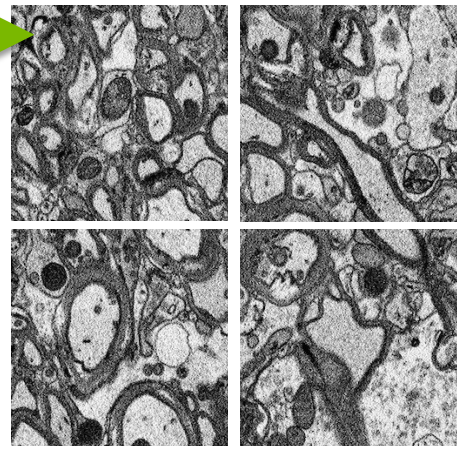


~1cm^3

~1mm^3

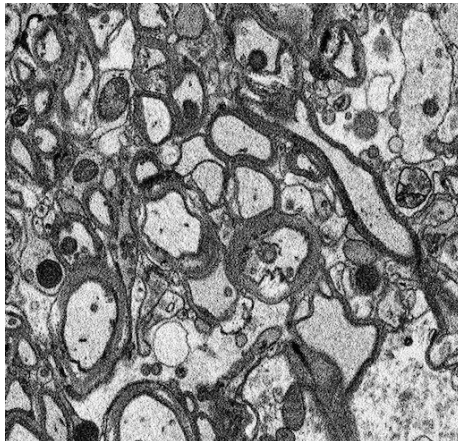


25000
40nm sections
1mm x 1mm
(6nm resolution)



Each section
imaged with EM as
N tiles (8 bit)

80K x 40K pixels



Sections
stitched
together

How much image data is 1mm^3 ? 1e15 voxels -> ~1 PB

DATA CHALLENGES IN CONNECTOMICS

Mouse brain: 70M neurons

~1cm^3

How much image data is 1cm^3 ? **~1EB**

Human brain: 80B neurons

~1000cm^3

How much image data is 1000cm^3 ? **~1000 EB**

(6nm x 6nm x 40nm)

Reconstructed data will be much larger:

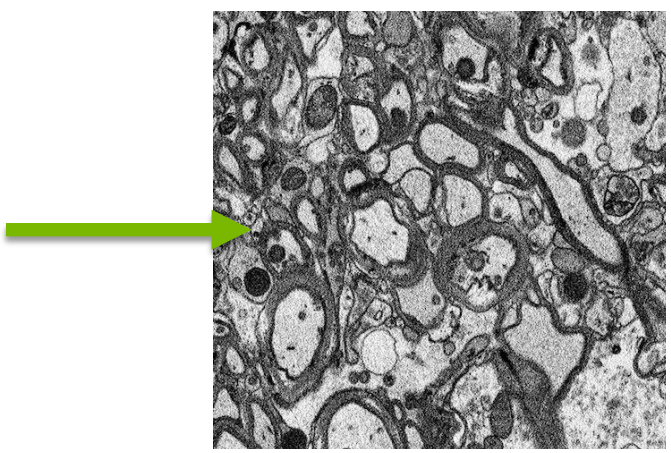
- **Segmentation labels for each voxel (4x voxel data)**
- **3D Mesh**
- **Skeleton**

The structures are expected to be used to seed simulations to study flow in neuro transmitters, in better modeling the brain, brain-inspired computing, among others.

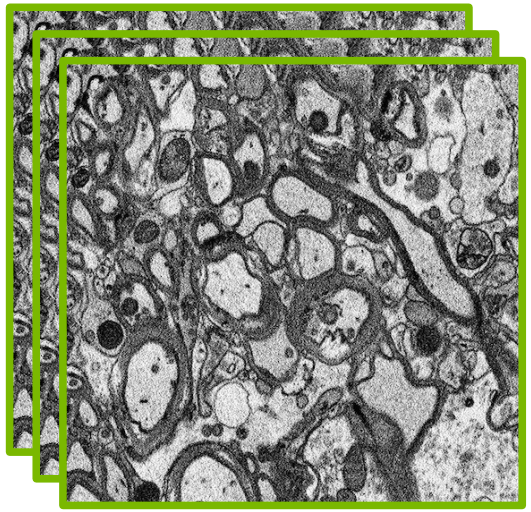
Courtesy: Nicola Ferrier, Tom Uram, Bobby Kasthuri (Argonne & UChicago)

Data from Gregg Wildenberg, Kasthuri Lab, UChicago

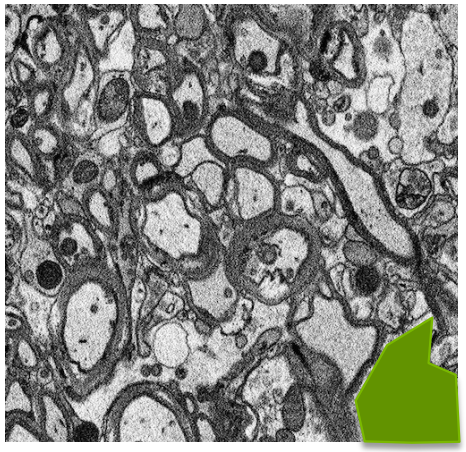
CONNECTOMICS PROCESSING



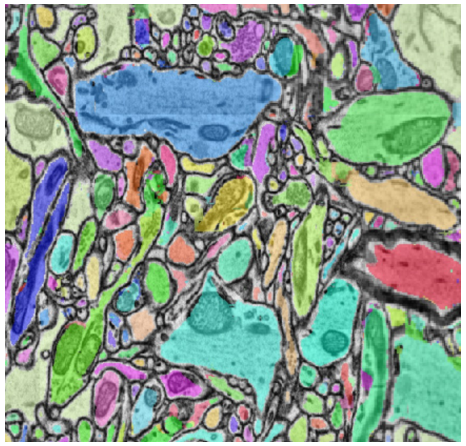
Sections stitched together



Align sections

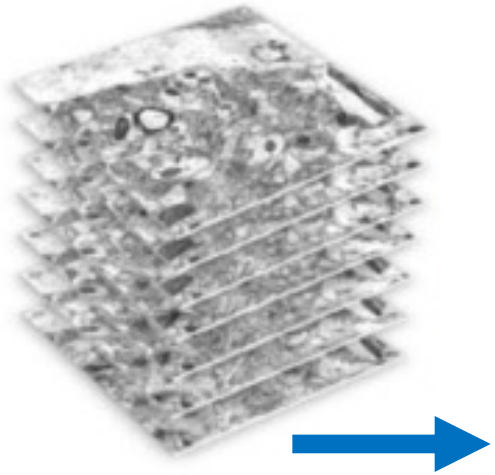


Mask out non-target objects

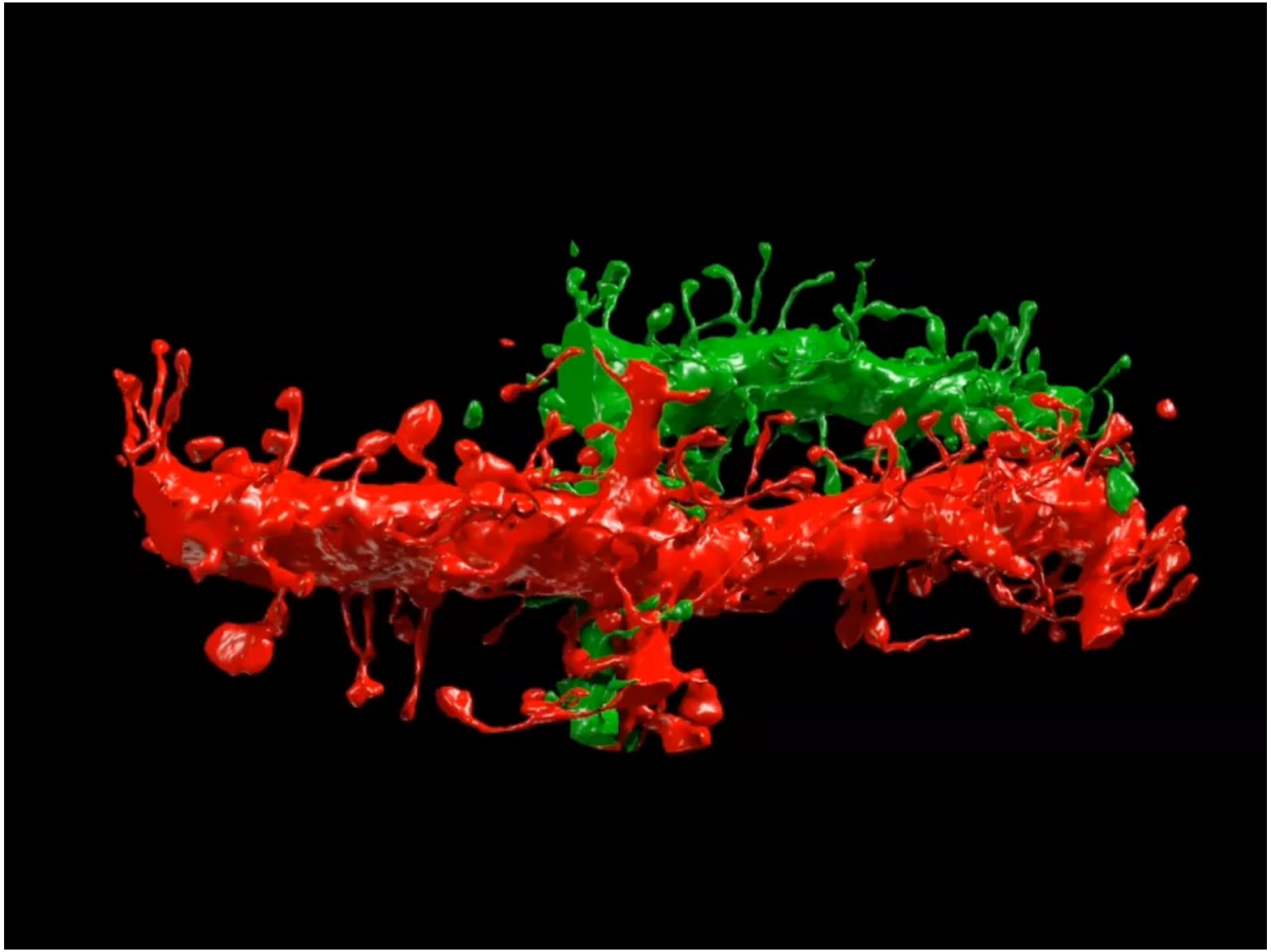


Segment target objects

RECONSTRUCTING THE BRAIN CONNECTIVITY



EM images



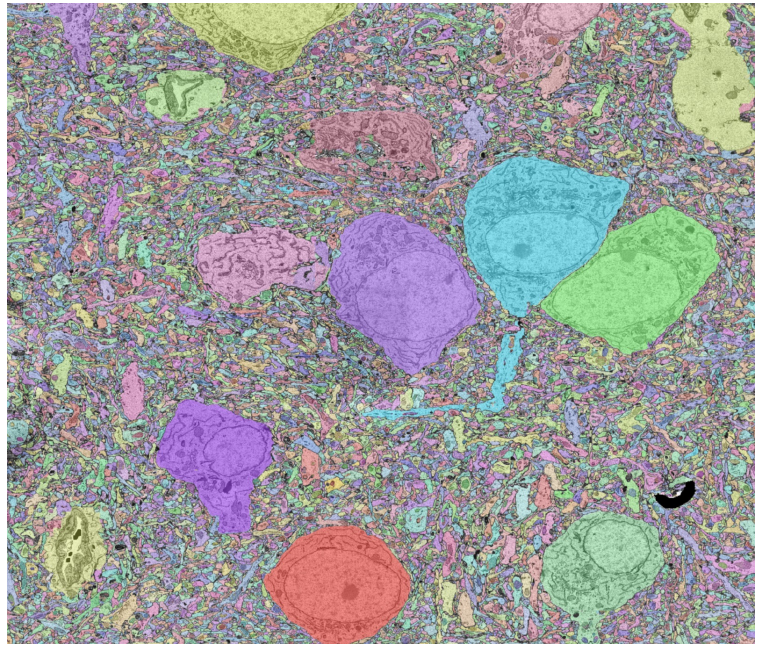
Kasthuri et al, Cell 2015

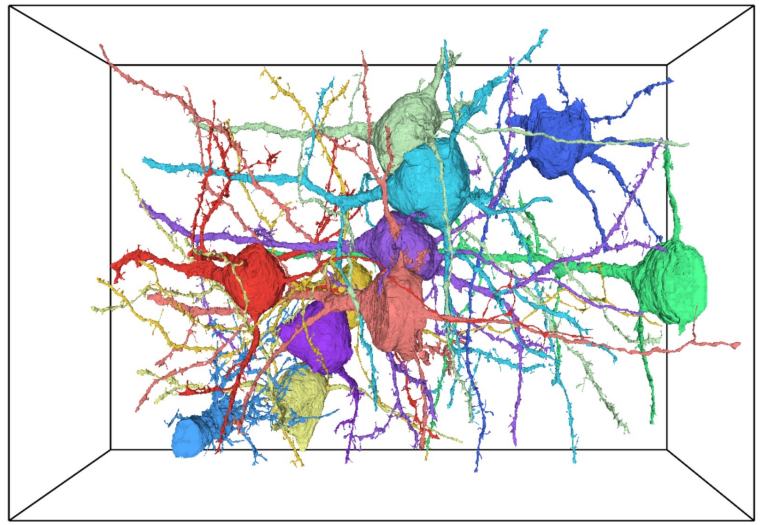
LARGE-SCALE RECONSTRUCTION

- **Inference (and training) has scaled** on CPU-based and GPU-based supercomputers (parallel granularity: overlapping subvolumes)
  - Achieved million-way concurrency on Theta supercomputer
- Image stitching and alignment components are being scaled as well to ensure a scalable end-to-end pipeline

Exascale Inference Problem:

- On a single GPU (A100), we achieve ~80 MegaVoxels/hour using 32-bit (There is still room for improvement here)
- In reduced precision (8-16 bits), we expect ~1 GigaVoxel/hour per GPU
- 1 PetaVoxel (1mm^3) will take ~1M GPU node hours
- Approximately, **24 hours on a system with 50K GPUs** (considering overlapping subvolumes)
- For a mouse brain (1cm^3), 1 ExaVoxel, we would need **~3 years on an exascale system**





Dong, et al, “Scaling Distributed Training of Flood-Filling Networks on HPC Infrastructure for Brain Mapping”, 2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS) at SC19

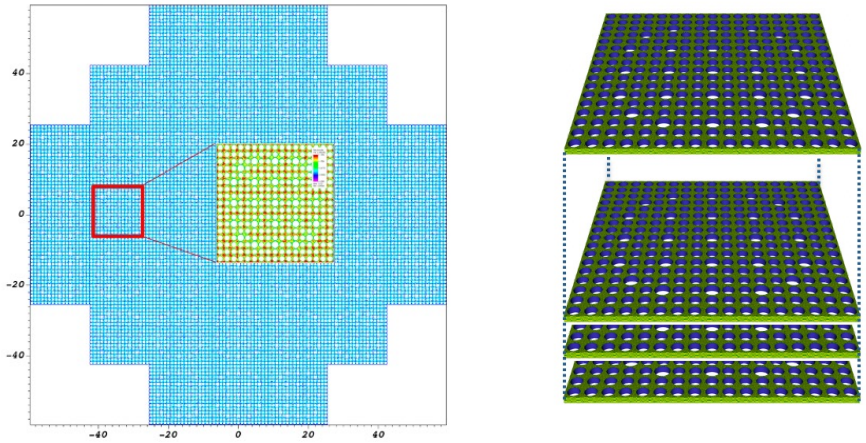
Vescovi, et al, “Toward an Automated HPC Pipeline for Processing Large Scale Electron Microscopy Data”, 2020 IEEE/ACM 2nd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP) at SC19

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

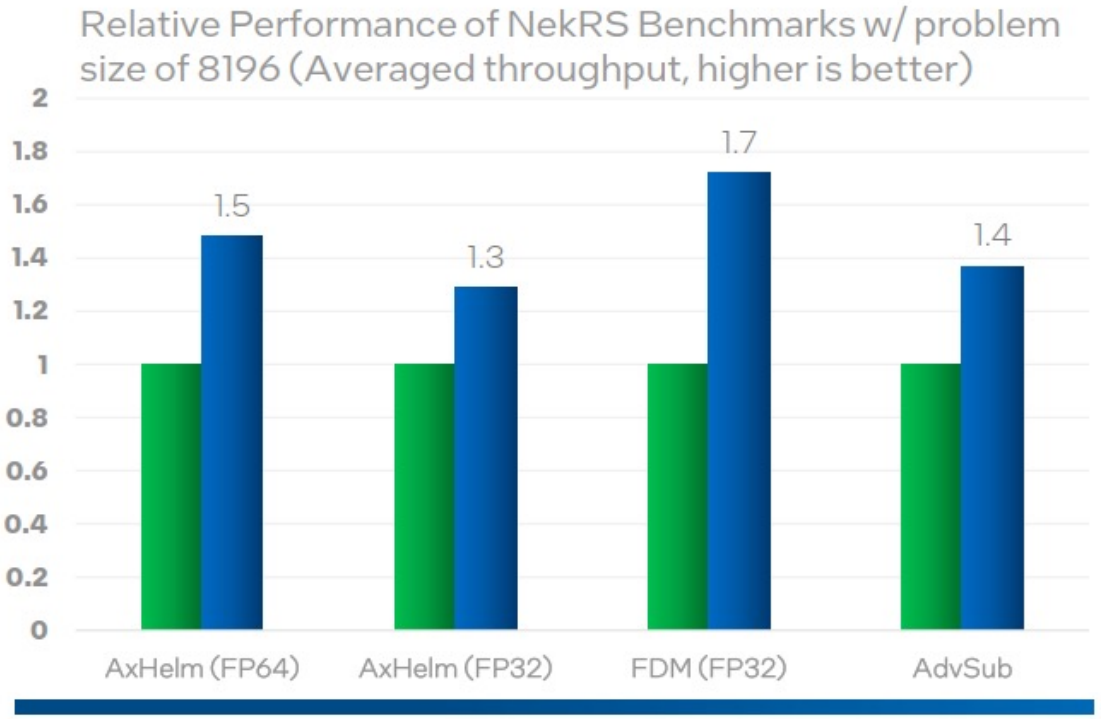
Intel® Data Center GPU Max Series
with Intel oneAPI DPC++ implementation

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.

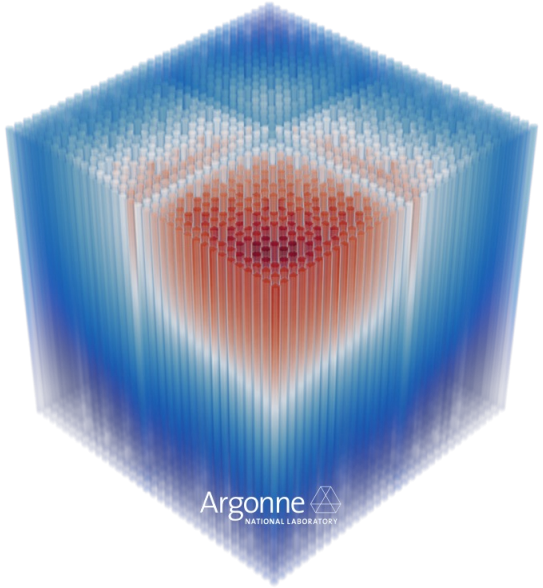
https://github.com/argonne-lcf/nekRS/

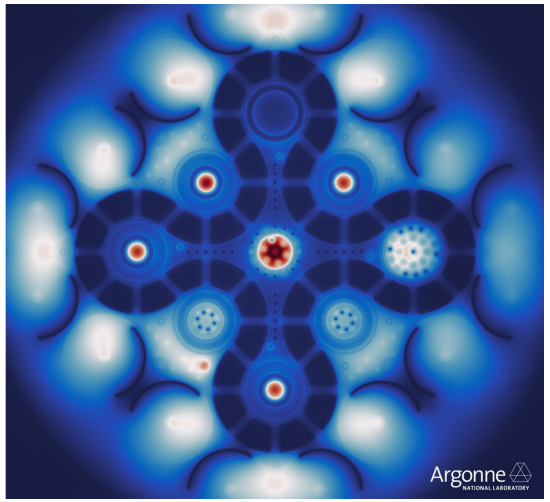
OCCA is an open-source library which aims to make it easy to program different types of devices (e.g. CPU, GPU, FPGA). It provides a unified API for interacting with backend device APIs (e.g. OpenMP, CUDA, OpenCL), uses just-in-time compilation to build backend kernel, and provide a kernel language,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

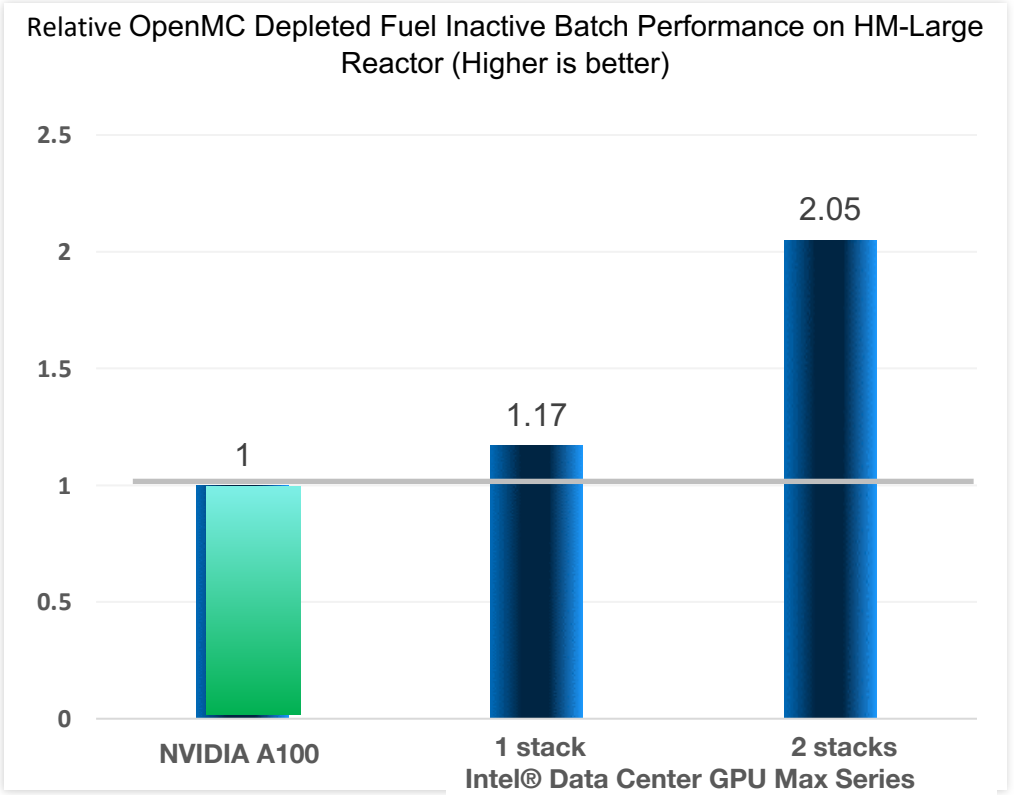
OpenMC performance

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/



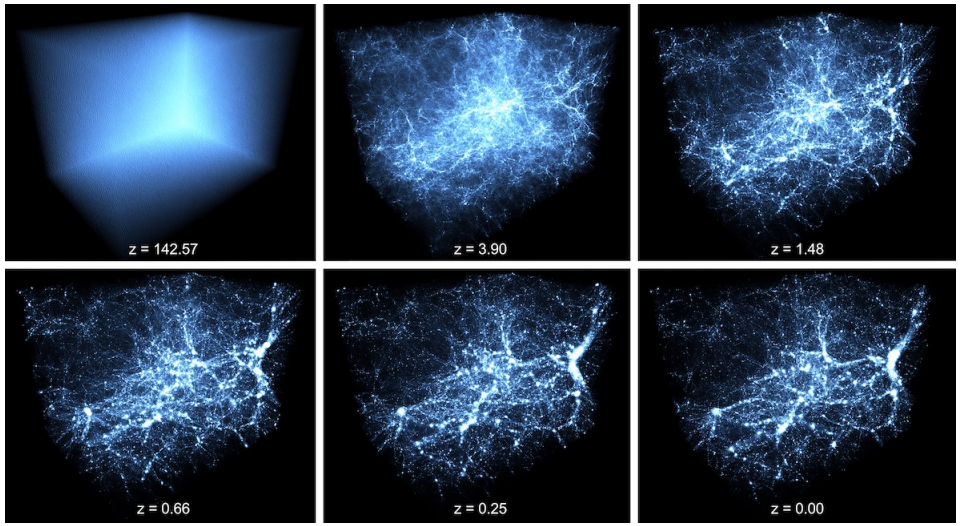




Near linear scaling from Intel® Data Center GPU Max Series 1 Stack to 2 Stack

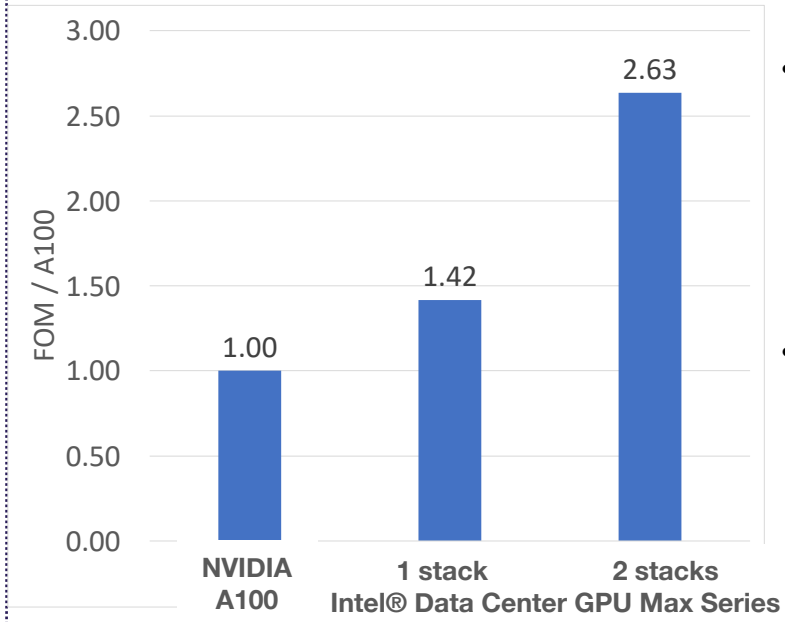
Application Summary: OpenMC is a Monte Carlo particle transport application that has recently been ported to the OpenMP target offloading programming model for use on GPU-based systems. The Monte Carlo method employed by OpenMC is considered the "gold standard" for high-fidelity simulation while also 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.

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

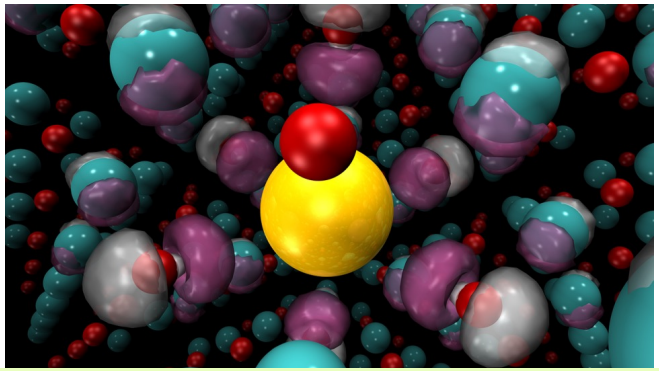


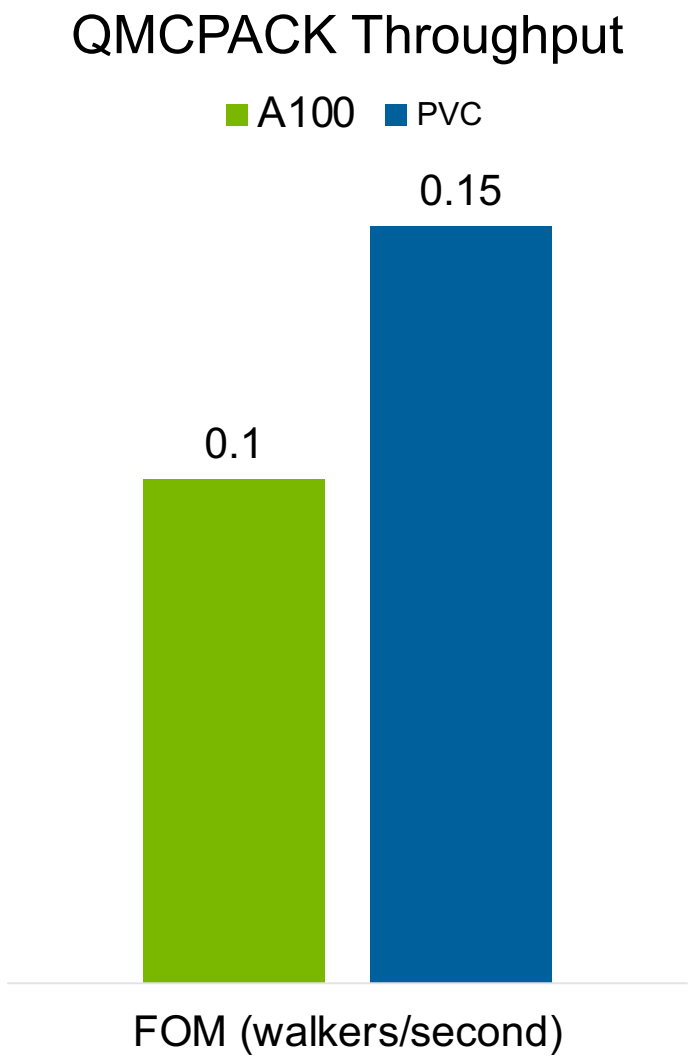
- Original CUDA kernels translated to SYCL using SYCLomatic, with the five most compute-intensive kernels hand-optimized by Intel performance engineers.
- Implemented optimizations included loop restructuring to take advantage of SYCL subgroup broadcast performance.

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.

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.
  - PVC: 2 MPI ranks, with one MPI rank, 8 Walkers, 64 GB of HBM per Tile. 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.



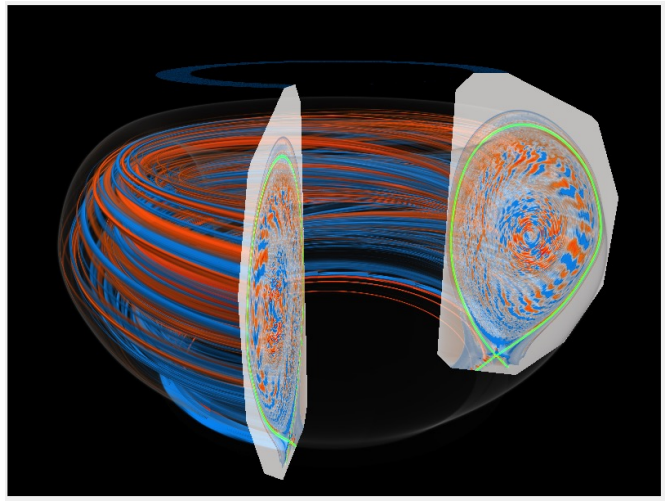


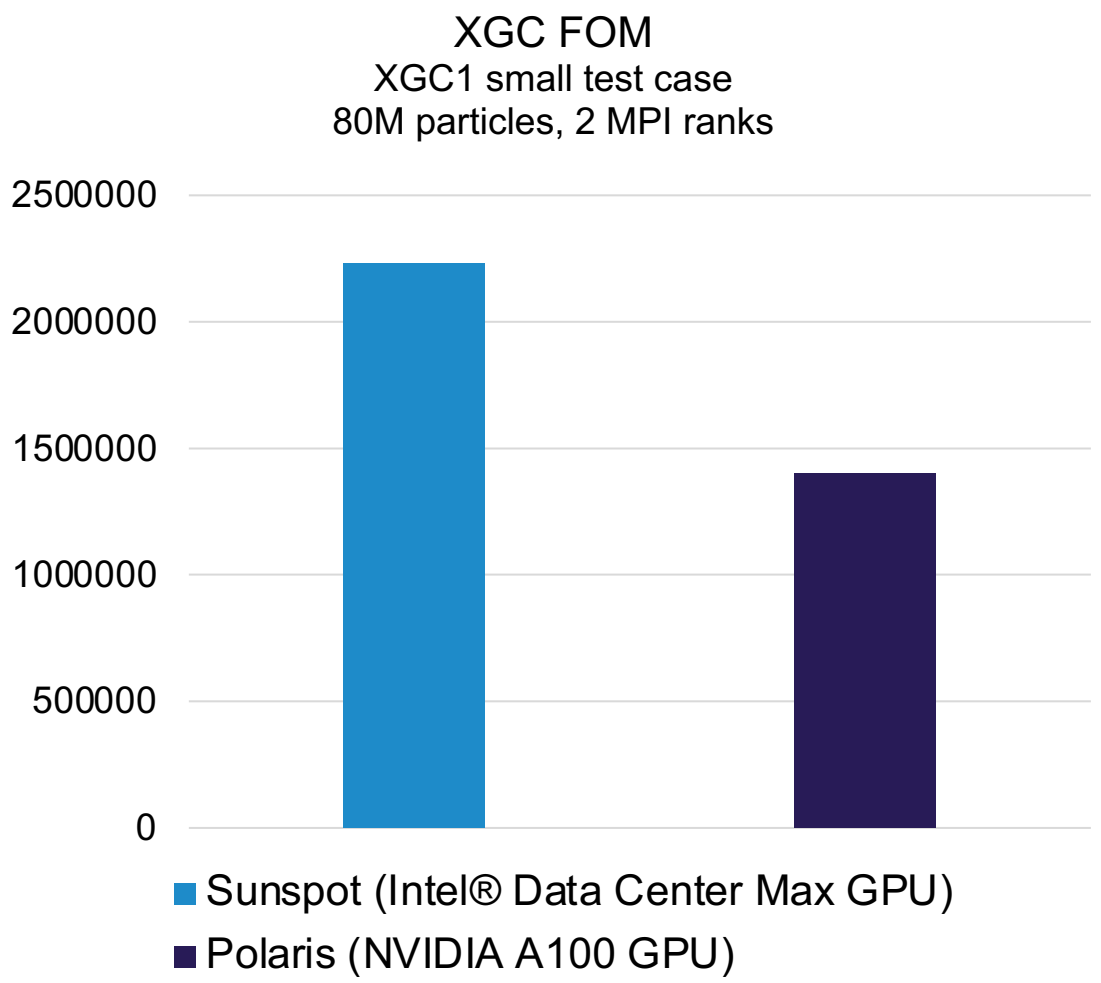
Argonne POCs: A. Benali, Y. Luo, T. Applencourt
Intel POC: Y. Kim

WDMApp: XGC Performance

ESP Project PI: CS Chang

- 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.





THANK YOU

www.anl.gov

Argonne NATIONAL LABORATORY