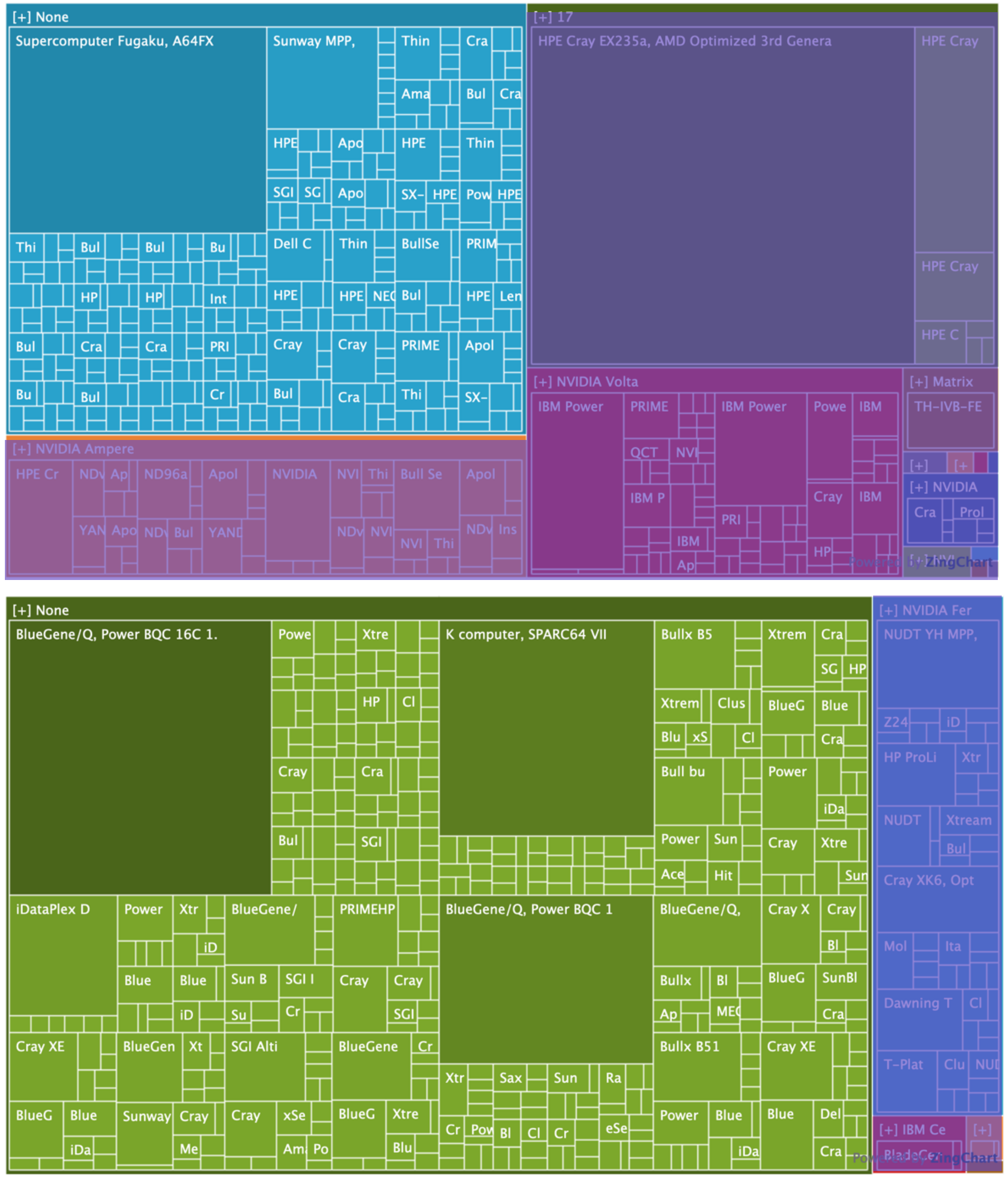
Aurora Exascale Architecture

Servesh Muralidharan
Computer Scientist, Performance Engineering Team
Argonne Leadership Computing Facility

PATH TO EXASCALE

Elements of a supercomputer

- Processor
  - —architecturally optimized to balance complexity, cost, performance, and power
- Memory
  - —generally commodity DDR, amount limited by cost
- Node
  - —may contain multiple processors, memory, and network interface
- Network
  - —optimized for latency, bandwidth, and cost
- IO System
  - —complex array of disks, servers, and network
- Software Stack
  - —compilers, libraries, tools, debuggers, …
- Control System
  - —job launcher, system management



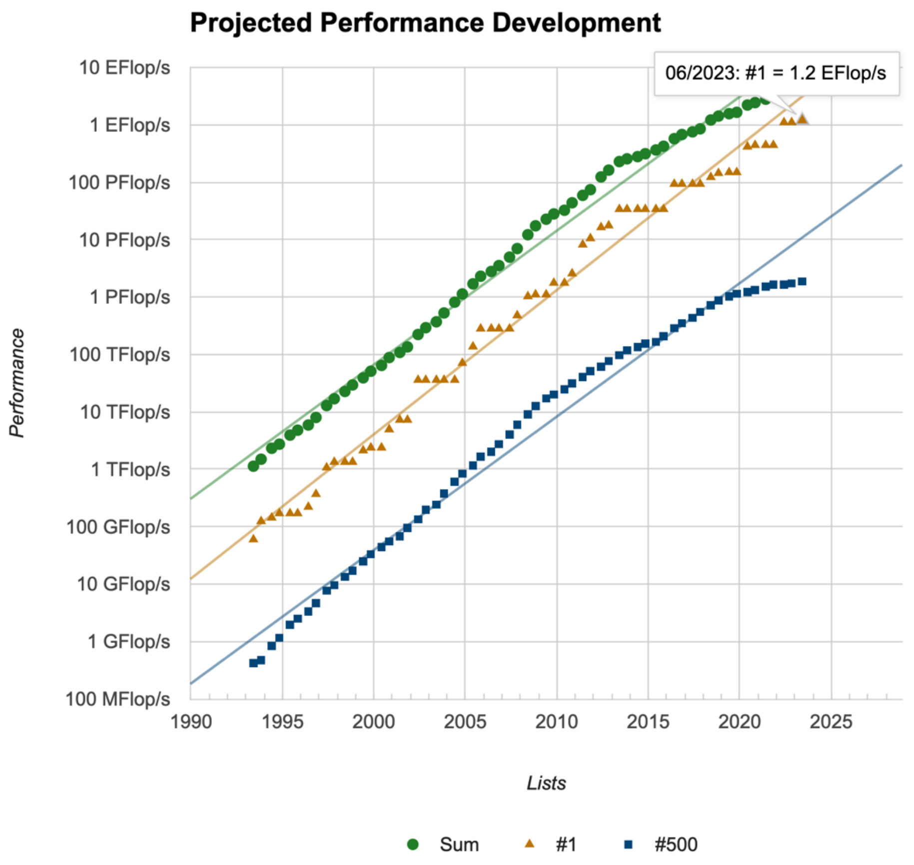
June 2022

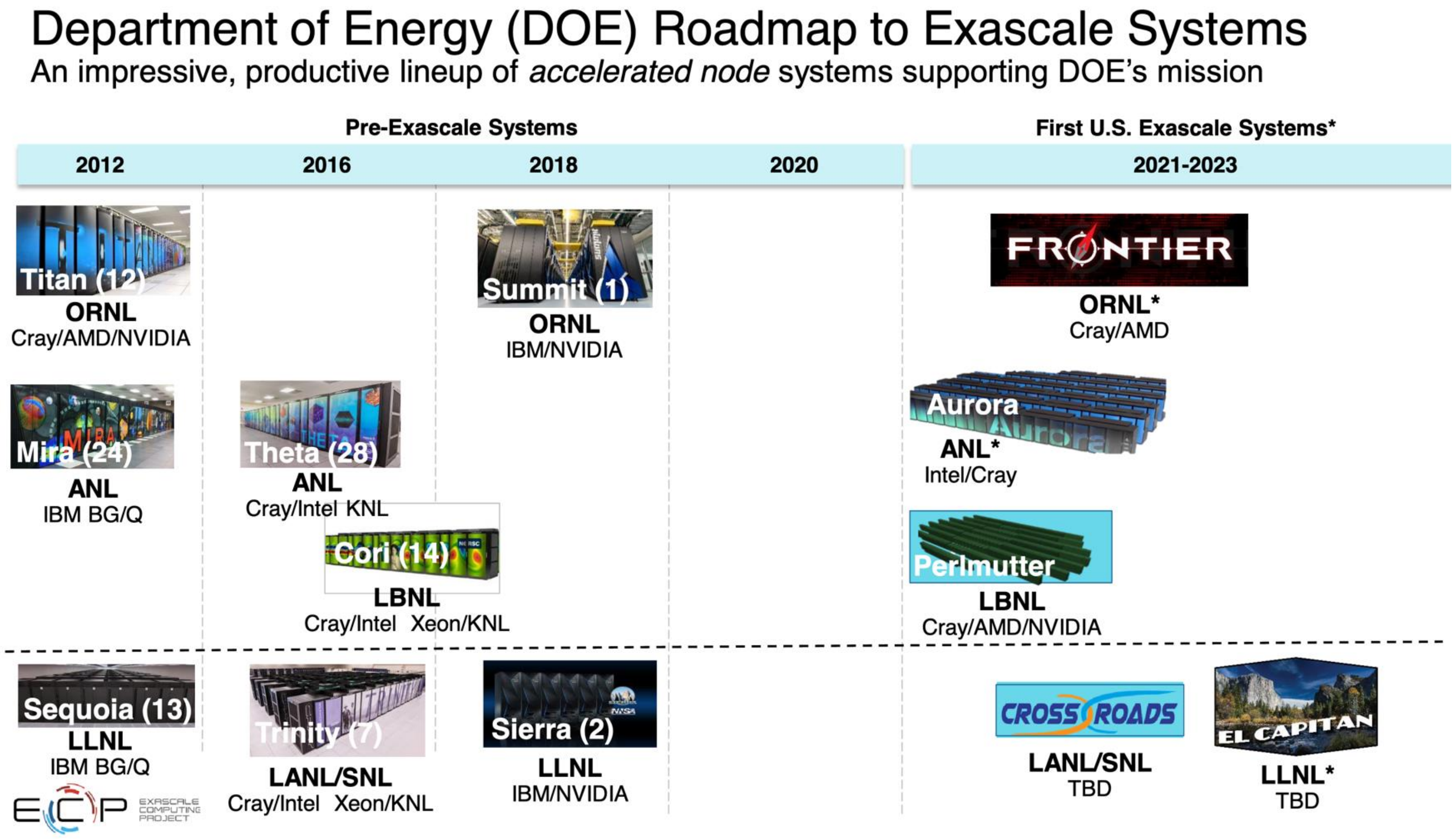
June 2012

https://www.top500.org/statistics/treemaps/

Argonne NATIONAL LABORATORY

Exascale Computing Project



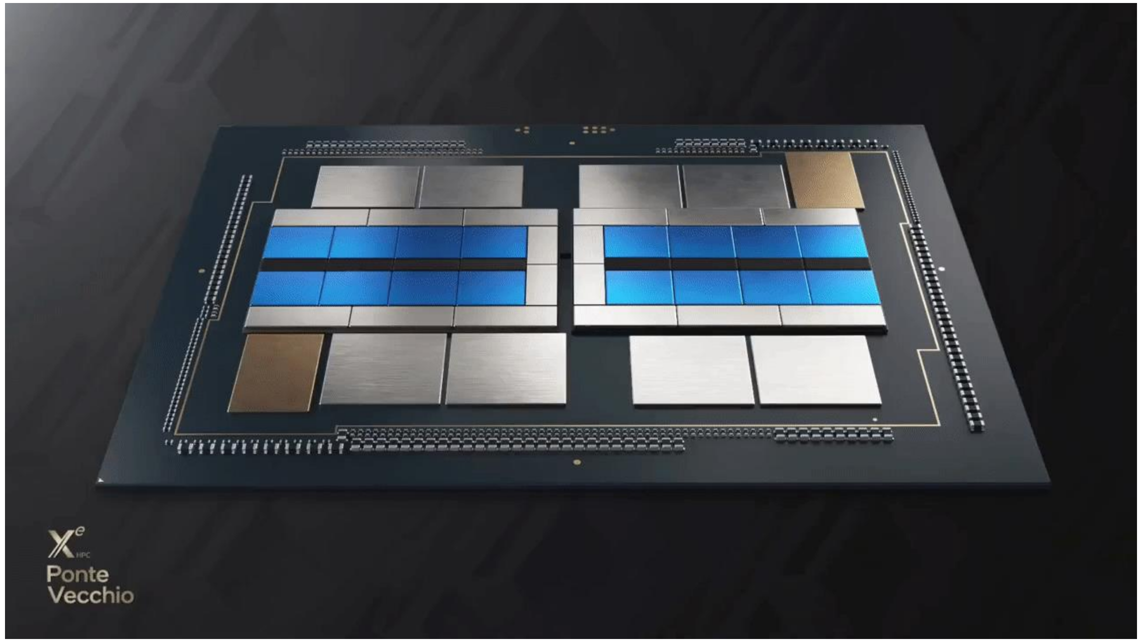


https://www.top500.org/statistics/perfdevel/

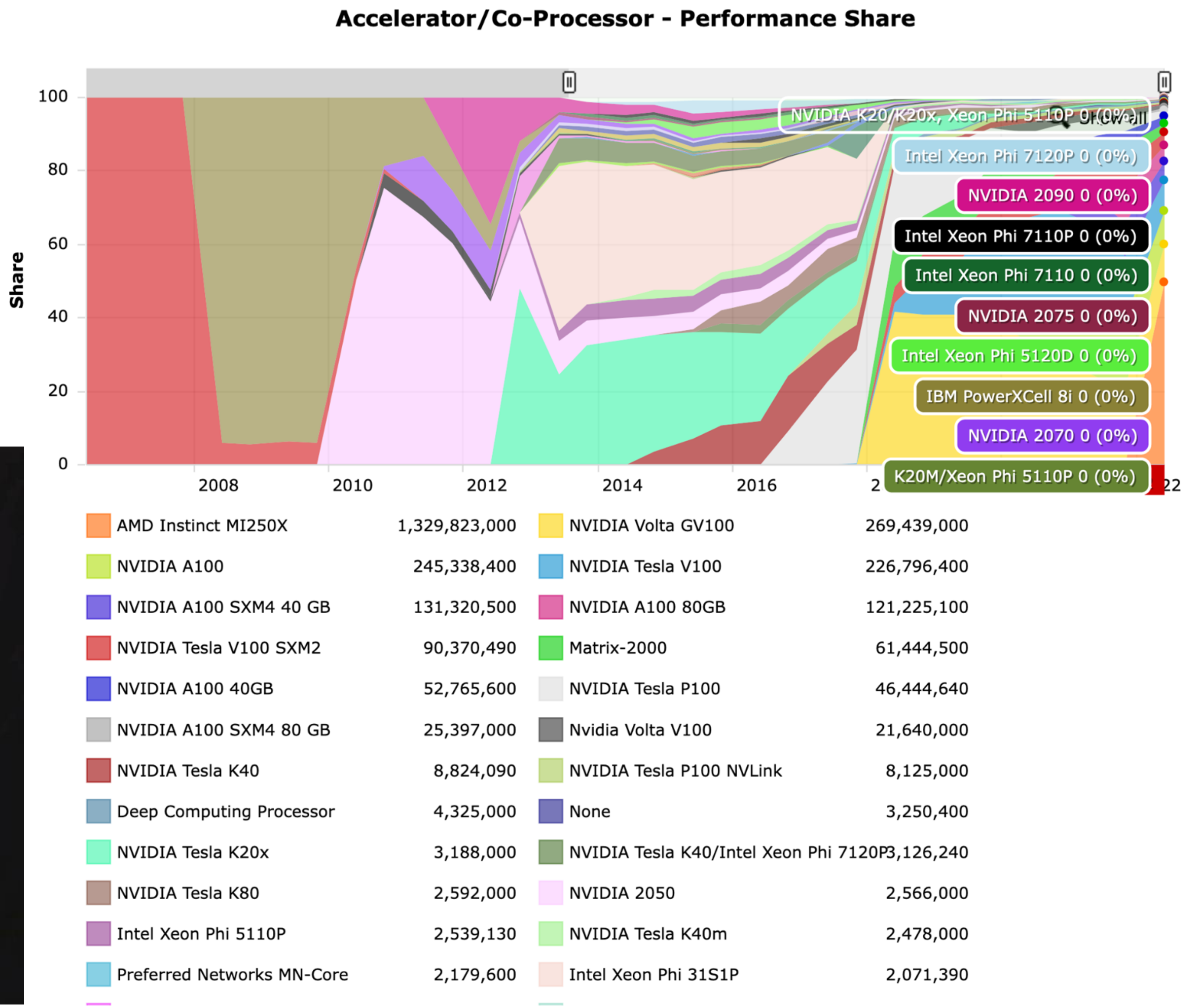
https://science.osti.gov/-/media/bes/besac/pdf/201907/1330_Diachin_ECP_Overview_BESAC_201907.pdf

Path to Exascale Computing

- Era of data parallel computing
  - — Dominated by GPUs
  - — Exploit SIMT/SIMD Parallelism
- Architectural Challenges
  - — Multichip Packaging
  - — Next generation technologies



Intel's HPC GM Trish Damkroger Keynote ISC 2021
https://www.youtube.com/watch?v=PuEcCRJLrvs
https://download.intel.com/newsroom/2021/data-center/Intel-ISC2021-keynote-presentation.pdf



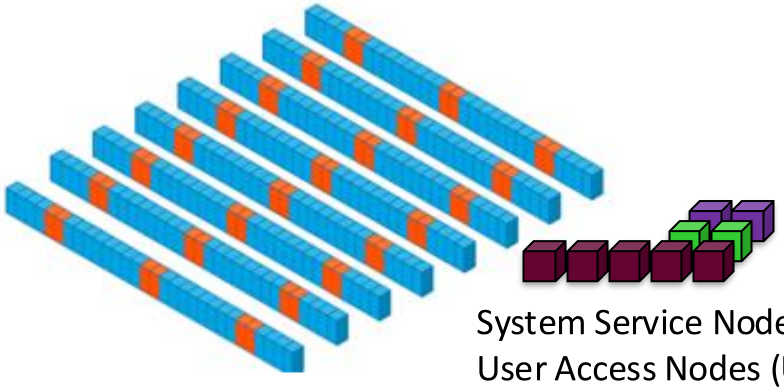
https://www.top500.org/statistics/overtime/

Argonne NATIONAL LABORATORY

AURORA: HARDWARE

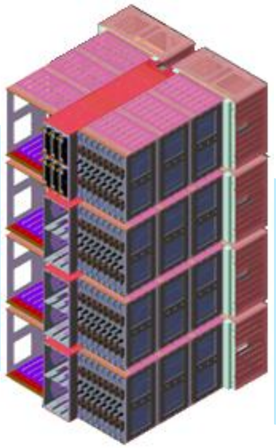
Argonne
NATIONAL LABORATORY

Aurora High-level System Overview

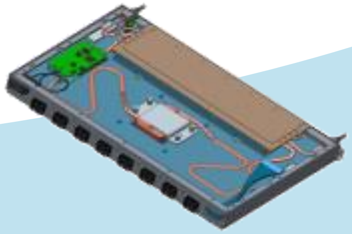


System Service Nodes (SSNs)
User Access Nodes (UANs)
DAOS Nodes (DNs)
Gateway Nodes (GNs)
IOF service, scalable library loading
DAOS <-> Lustre data mover

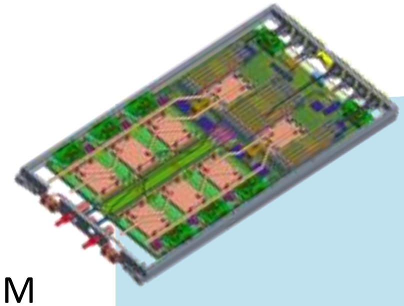
AURORA SYSTEM
166 Compute racks
10,624 Nodes
GPU: 8.16 PB HBM
CPU: 1.36 PB HBM, 10.9 PB DDR5
DAOS: 64 racks, 1024 nodes
230 PB (usable), 31 TB/s



COMPUTE RACK
64 Compute blades
32 Switch blades
GPU: 49.1 TB HBM
CPU: 8.2 TB HBM, 64 TB DDR5



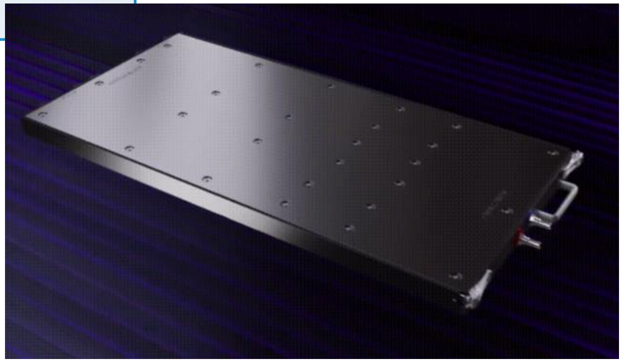
SWITCH BLADE
1 Slingshot switch
64 ports
Dragonfly topology

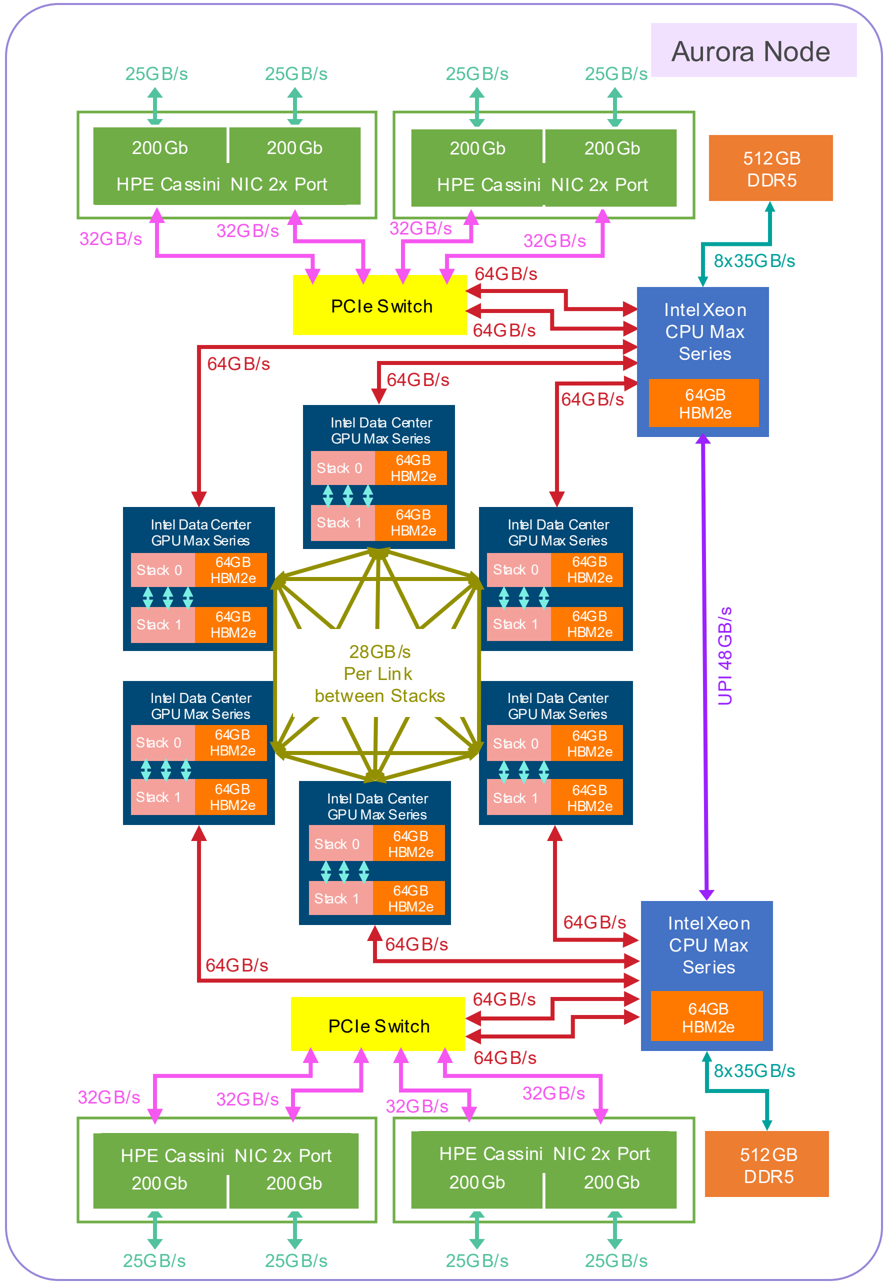


COMPUTE BLADE
2x Intel Xeon Max Series w HBM
6x Intel Data Center GPU Max Series
GPU: 768 GB HBM
CPU: 128 GB HBM, 1024 GB DDR5

Aurora Exascale Compute Blade

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

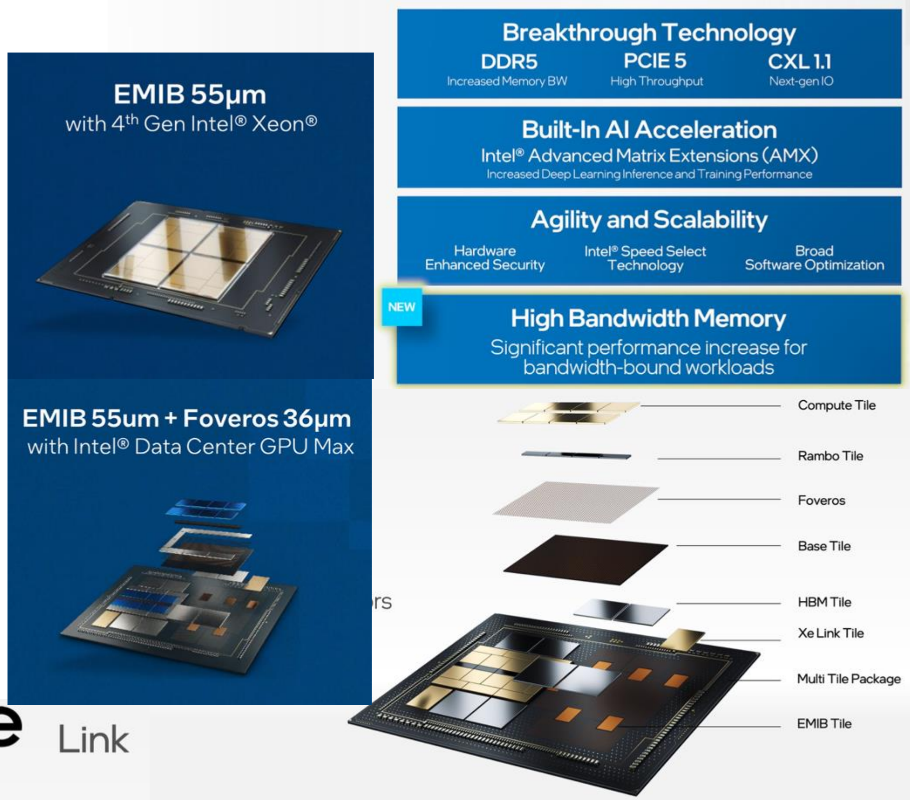




Aurora Exascale Compute Blade - Components

- Intel Xeon Max Series CPU w HBM
  - DDR5 and HBM
  - PCIe Gen5
- Intel Data Center Max Series GPU
  - Multi Tile architecture
    - Compute Tile
      - Xe Cores
      - L1 Cache
    - Base Tile
      - PCIe Gen5
      - HBM2e Main Memory
      - MDFI
      - EMIB
- GPU – GPU Interconnect
  - Xe Link

https://download.intel.com/newsroom/2021/data-center/Intel-ISC2021-keynote-presentation.pdf



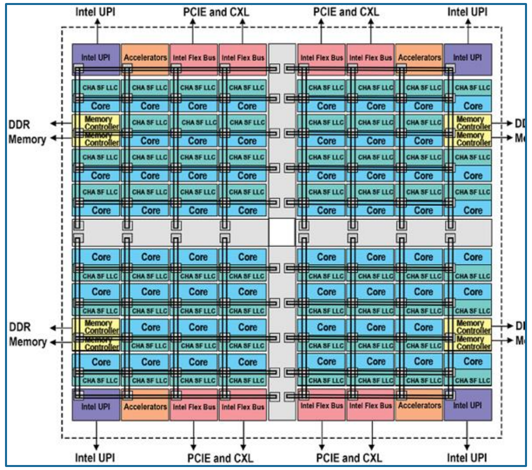
https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html

Argonne NATIONAL LABORATORY

Intel Xeon Max Series CPU w HBM

- Dual socket
- 52 cores
- First Level Cache: 32 KB Instruction Cache
  48 KB Data Cache
- Mid-Level Cache: 2 MB private per core
- Last Level Cache: 1.875 MB per core
- 8 channels DDR5 @ 4400MT/s
- 1TB DDR5 Memory
- 64GB HBM2e per socket
- 80 PCIe lanes with PCIe Gen 5.0 support
  - PCIe bifurcation support: x16, x8, x4, x2(Gen4)

https://www.hc33.hotchips.org/assets/program/conference/day1/HC2021.C1.4%20Intel%20Arijit.pdf

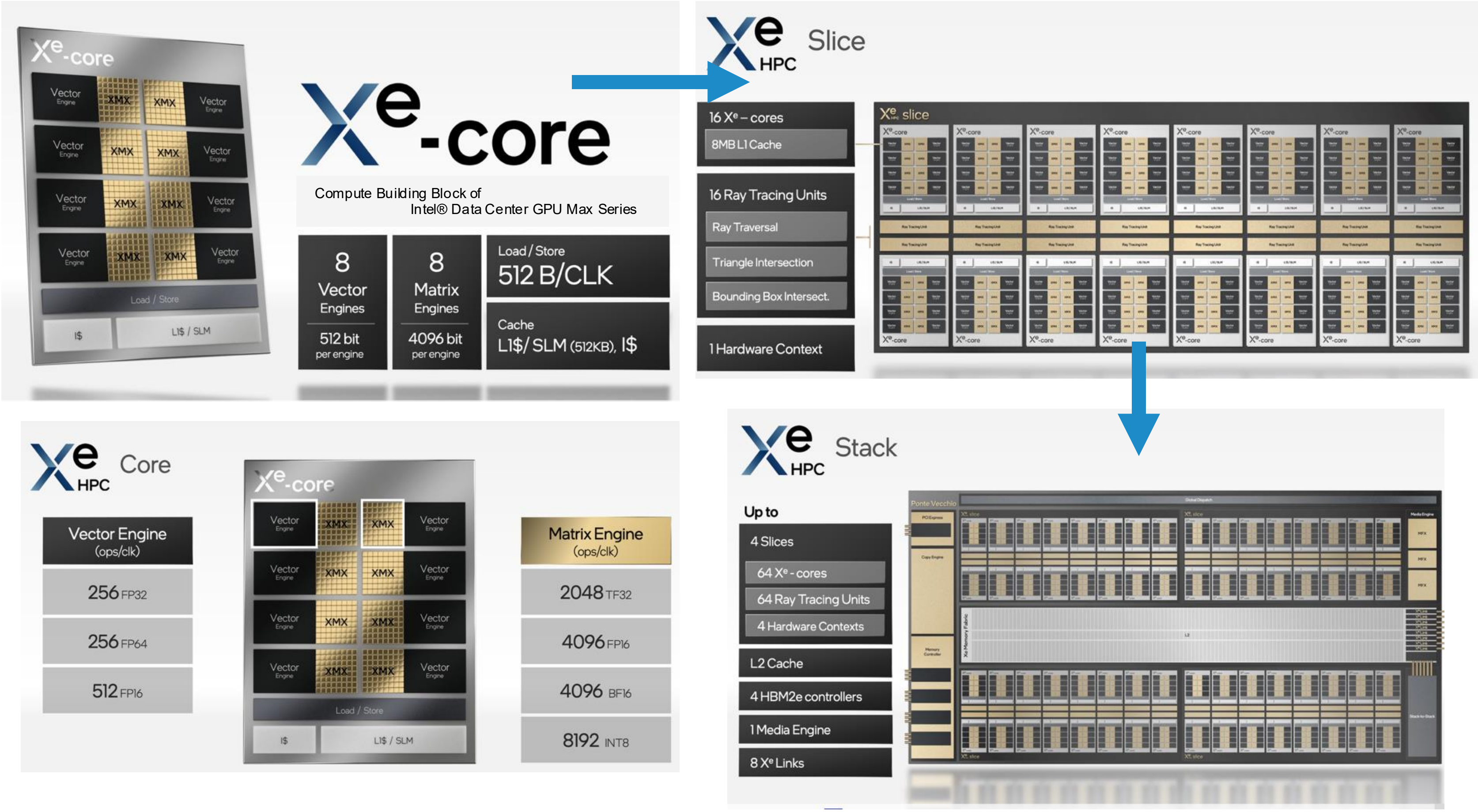


https://www.intel.com/content/www/us/en/developer/articles/technical/fourth-generation-xeon-scalable-family-overview.html

Argonne NATIONAL LABORATORY

Intel Data Center GPU Max Series Architectural Components

- Xe Cores
  - Vector Engine
    - Traditional compute pipeline
  - Matrix Engine
    - Low precision systolic pipeline
  - L1 Data Cache
    - Shared Local Memory
  - Instruction Cache
- Xe Slice
  - Hardware Context
  - Offload Units
- Xe Stack
  - LLC
  - HBM2e controllers
  - Xe link
  - Cache Memory Fabric
  - PCIe Endpoint
  - Hardware specific engines
  - Stack to Stack Interconnect
  - Xe links
    - Multi GPU Interconnect

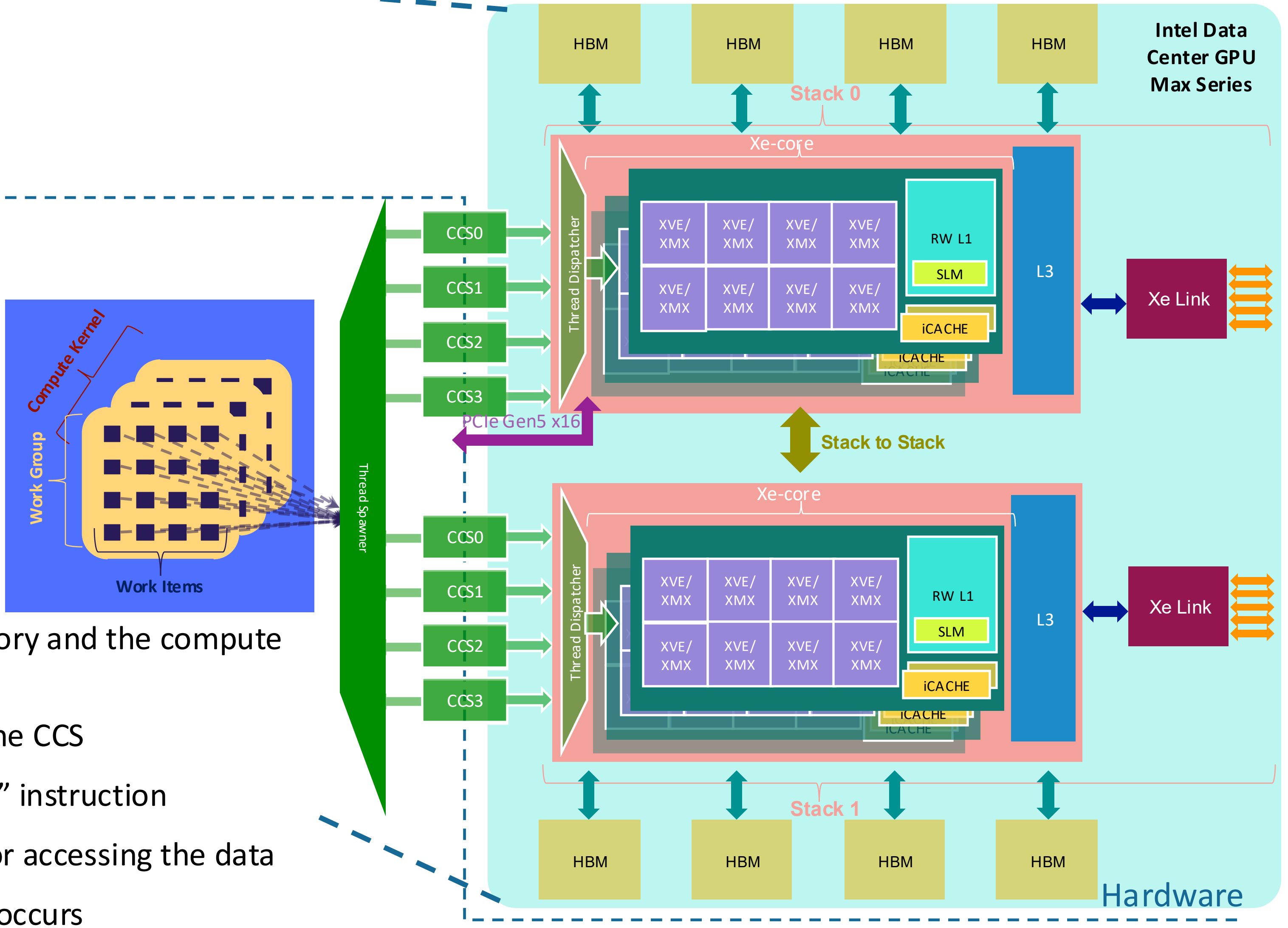


https://hc33.hotchips.org/assets/program/conference/day2/hc2021_pvc_final.pdf

GPU Compute Execution

XVE – Xe Vector Engine
GRF – General Register File
SLM – Shared Local Memory
RW L1 – Read/Write L1
HBM – High Bandwidth Memory
iCACHE – Instruction Cache
CCS – Compute Command Streamer
SIMD – Single Instruction Multiple Data

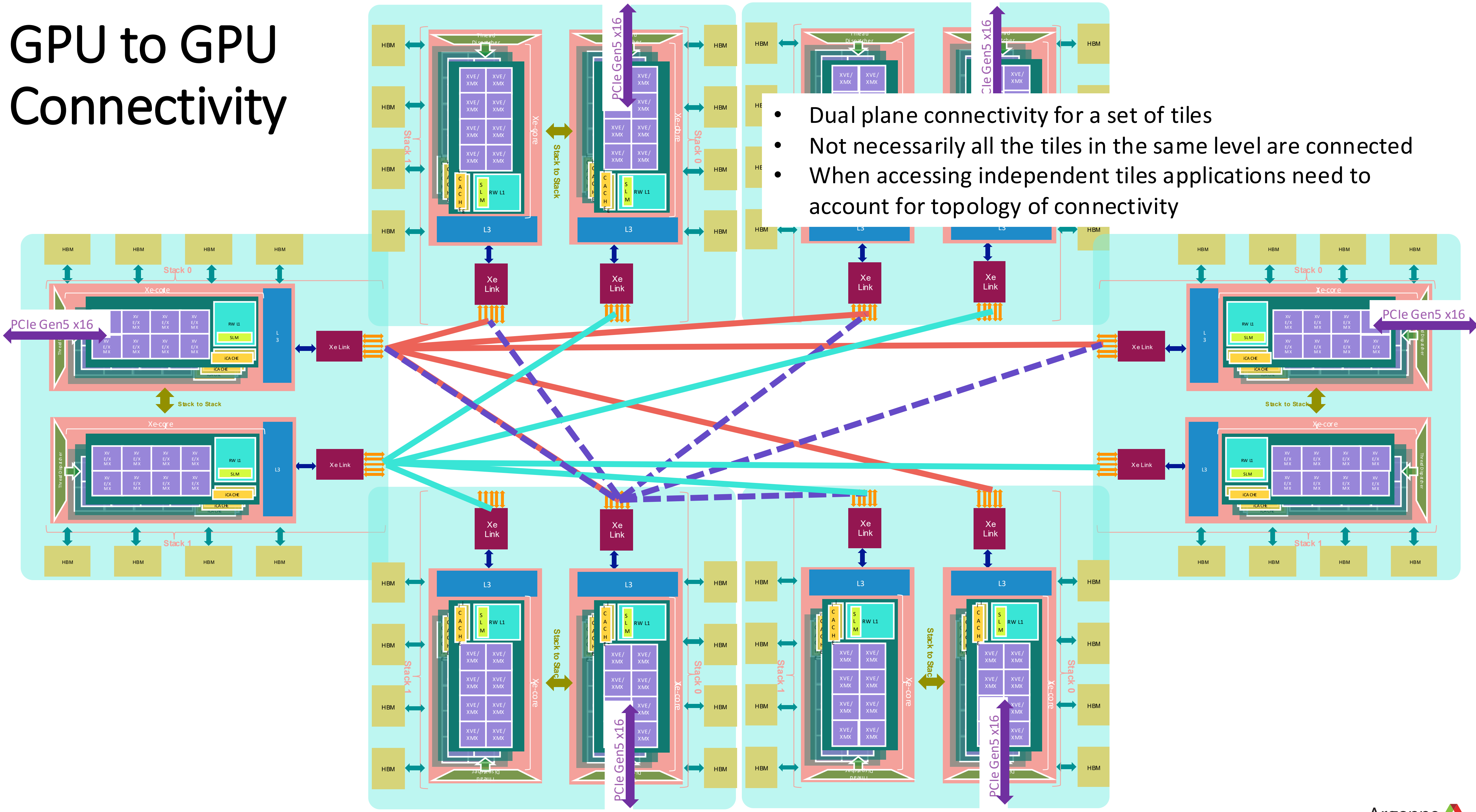
https://www.intel.com/content/www/us/en/docs/oneapi/optimization-guide-gpu/2024-0/intel-xe-gpu-architecture.html



https://www.intel.com/content/www/us/en/docs/oneapi/optimization-guide-gpu/2024-0/execution-model-overview.html

- Execution on the GPU starts with the allocation of memory and the compute kernel scheduled on the GPU
- The GPU threads are spawned and scheduled through the CCS
- Execution stops when the kernel hits the “end of thread” instruction
- Shared vs Device allocation implies different latencies for accessing the data
- GPU threads can switch when any of the stall condition occurs
  - However during execution threads cannot be interrupted

GPU to GPU Connectivity



- Dual plane connectivity for a set of tiles
- Not necessarily all the tiles in the same level are connected
- When accessing independent tiles applications need to account for topology of connectivity

Network Switch

Consistent, Repeatable Application Performance

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

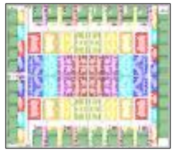
Extremely Scalable RDMA Performance

- 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



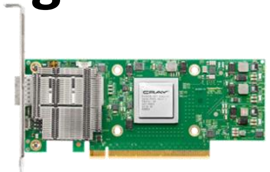
HPE Switch ASIC

Rack switches

100% DLC Switches

HPE Slingshot NICs - 200 Gbps

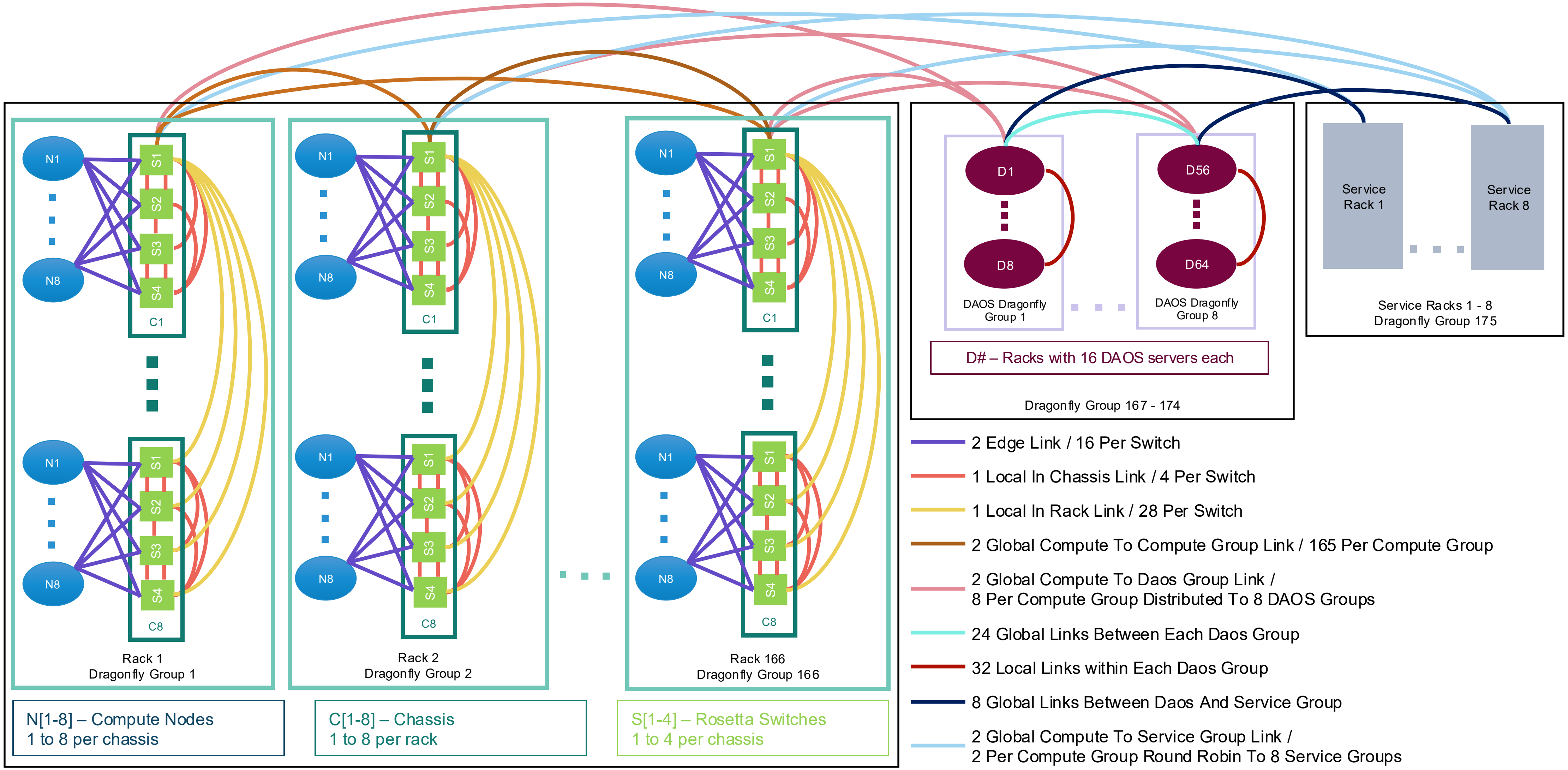
HPE NIC ASIC



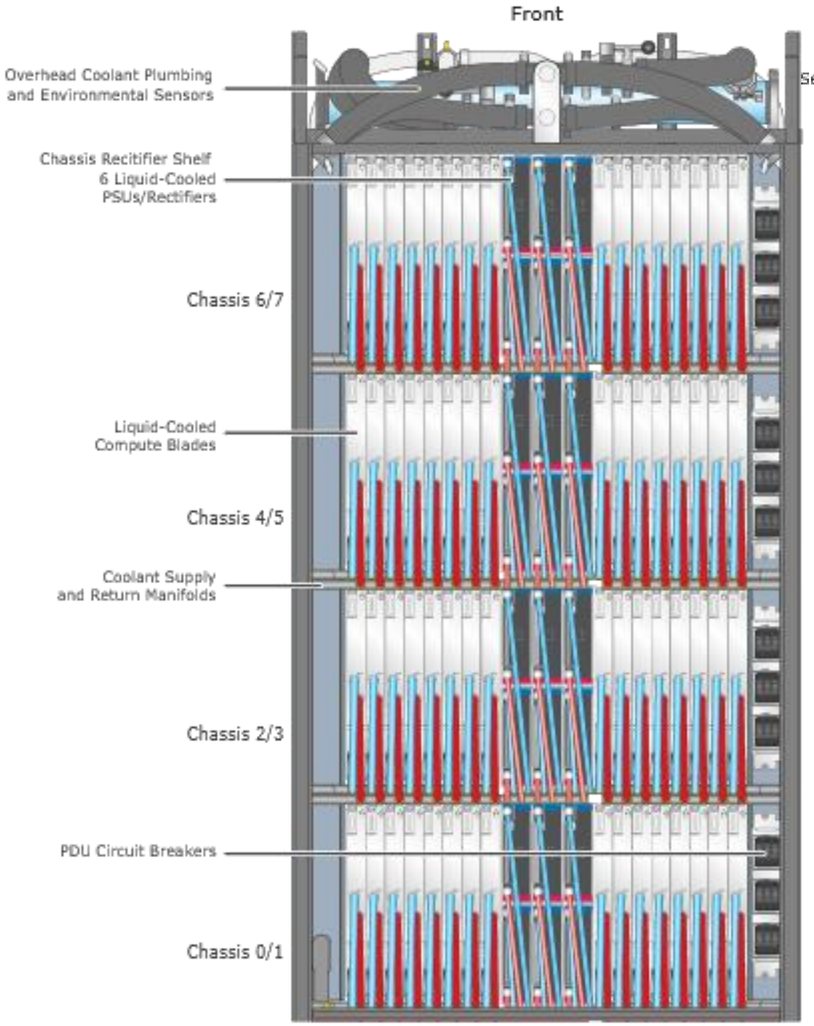
PCIe Adapters

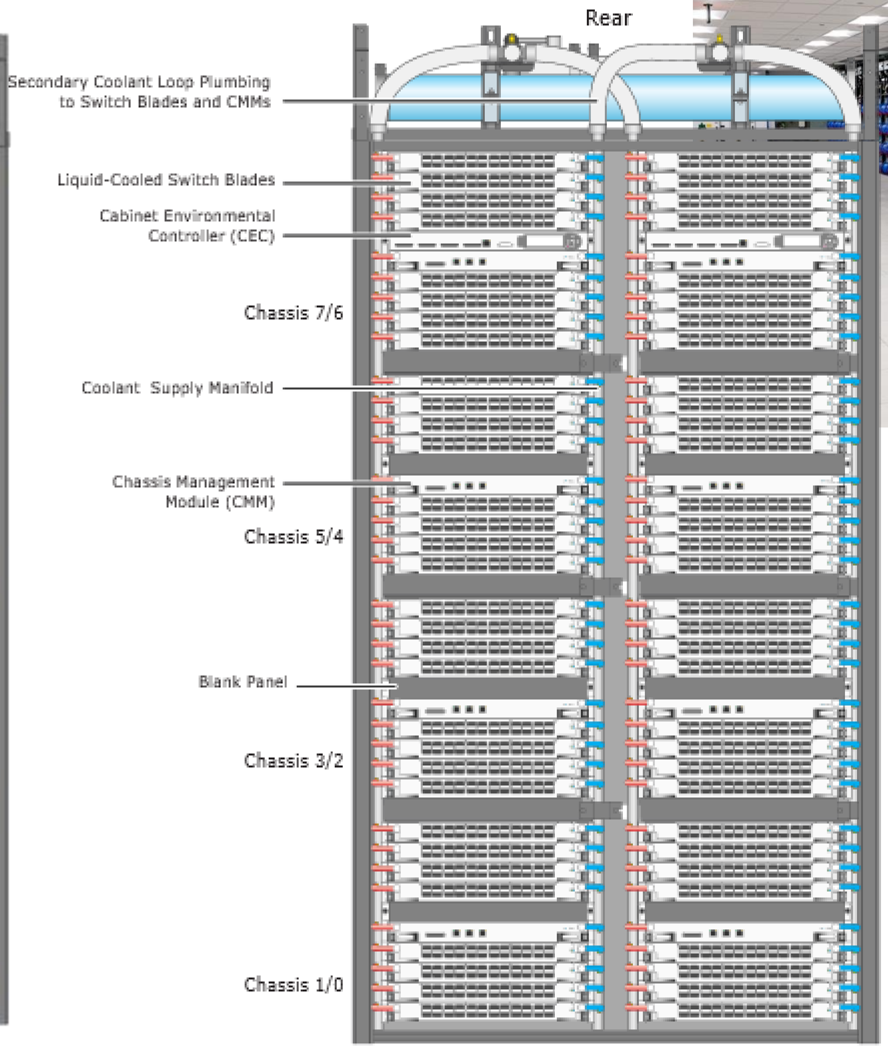
100% DLC NIC Mezz

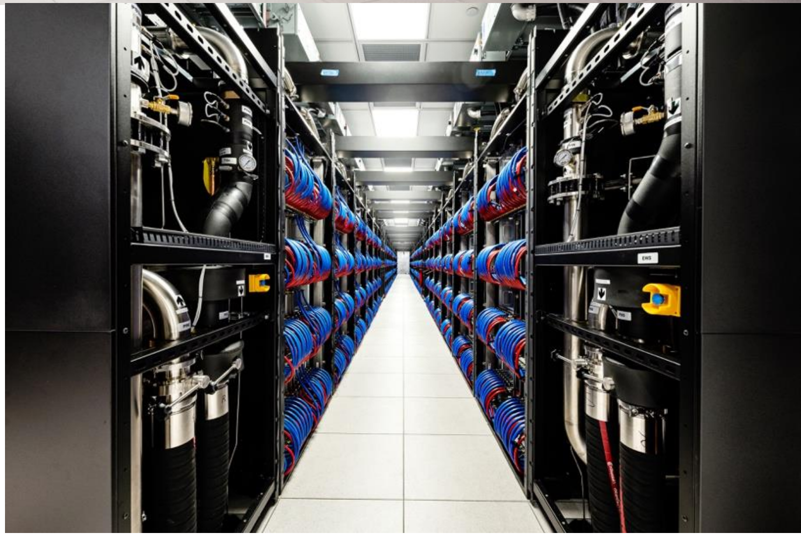
Aurora Dragonfly Interconnect



Aurora Cabinets at Argonne



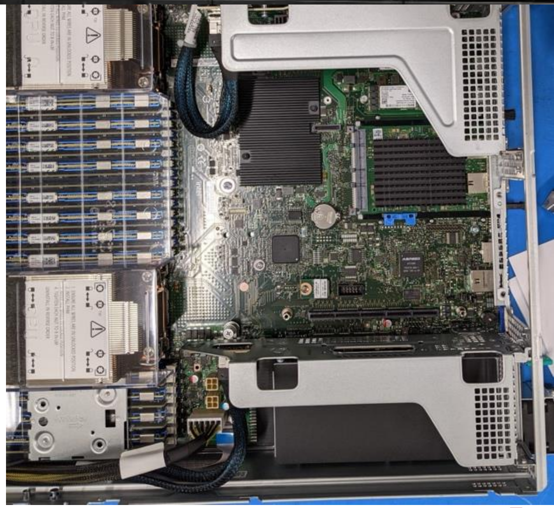




Aurora Storage Systems

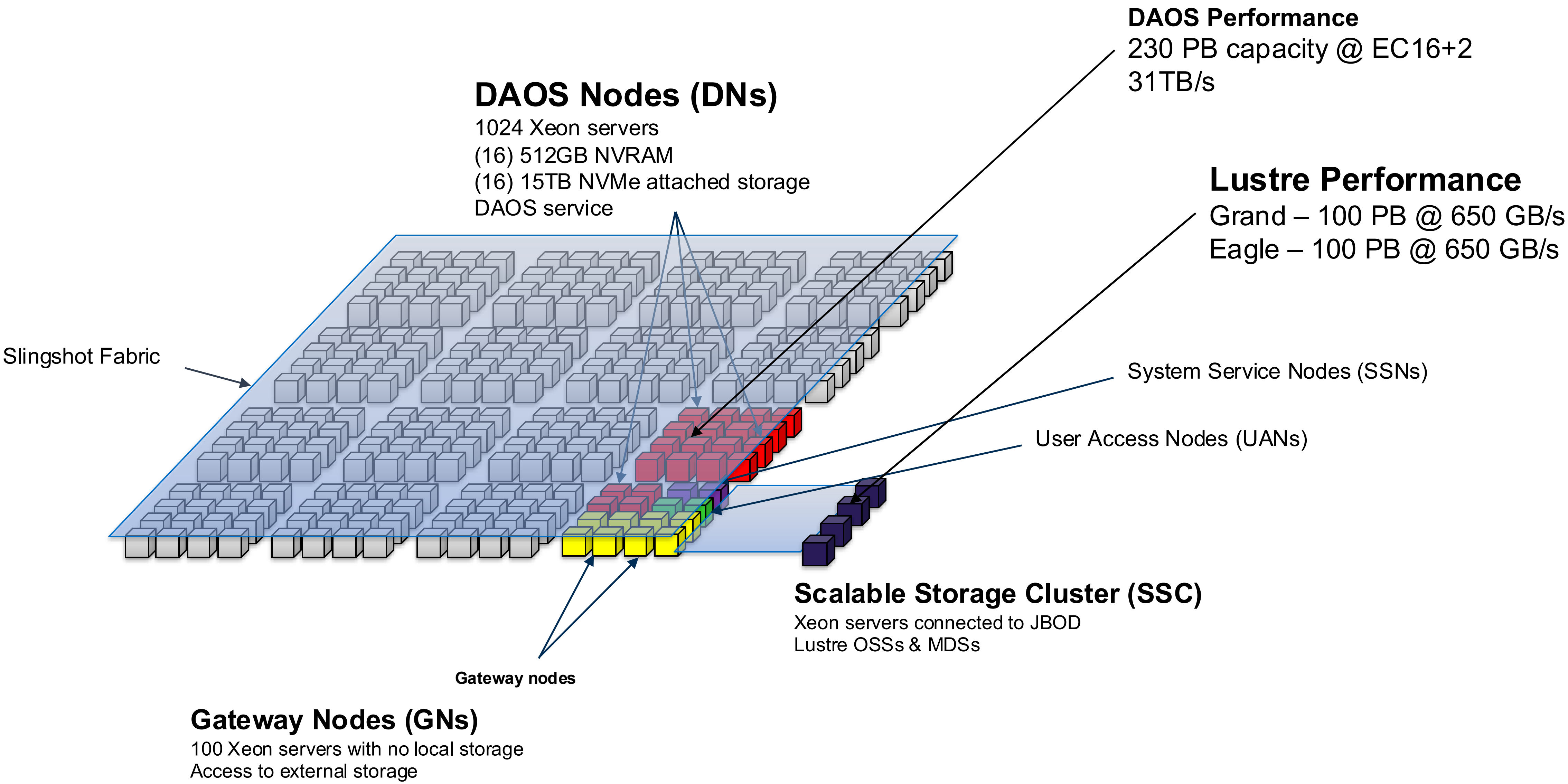
- DAOS provides Aurora's main "platform" high performance storage system
- Aurora leverages existing Lustre storage systems, Grand and Eagle, for center-wide data access and data sharing

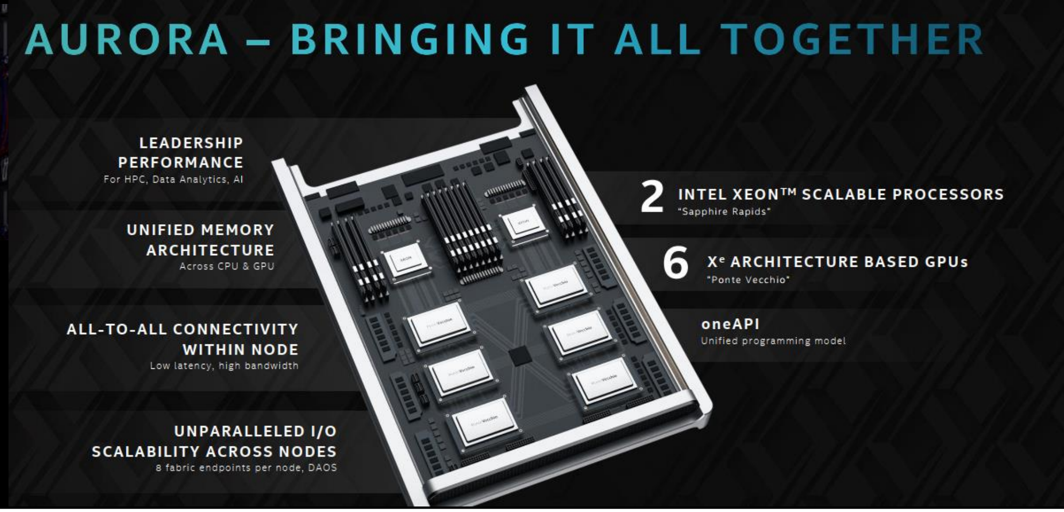
System	Capacity	Performance
Aurora DAOS	230 PB @ EC16+2 ▪ 250 PB NVMe ▪ 8 PB Optane PMEM	31 TB/s Read & Write
Eagle	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write
Grand	100 PB @ RAID6 ▪ 8480 HDD ▪ 40 Lustre MDT	> 650 GB/s Read & Write



- Intel Coyote Pass System
  - (2) Xeon 5320 CPU (Ice Lake)
  - (16) 32GB DDR4 DIMMs
  - (16) 512GB Intel Optane Persistent Memory 200
  - (16) 15.3TB Samsung PM1733
  - (2) HPE Slingshot NIC
- 1024 Total Servers
  - Each node will run 2 DAOS engines
  - 2048 DAOS engines

Aurora Storage Overview





Peak Performance
≧ 2 Exaflops DP

Intel GPU
Intel® Data Center GPU
Max Series 1550

Intel Xeon Processor
Intel® Xeon Max Series 9470C
CPU with High Bandwidth
Memory

Platform
HPE Cray-Ex

Compute Node
2x Intel® Xeon Max Series processors
6x Intel® Data Center GPU Max Series
8x Slingshot11 fabric endpoints

GPU Architecture
Intel XeHPC architecture
High Bandwidth Memory

Node Performance
>130 TF

System Size
166 Cabinets
10,624 Nodes
21,248 CPUs
63,744 GPUs

System Memory
1.36PB HBM CPU Capacity
10.9PB DDR5 Capacity
8.16PB HBM GPU Capacity

System Memory Bandwidth
30.58PB/s Peak HBM BW CPU
5.95PB/s Peak DDR5 BW
208.9PB/s Peak HBM BW GPU

High-Performance Storage
230PB
31TB/s DAOS bandwidth
1024 DAOS Nodes

System Interconnect
HPE Slingshot 11
Dragonfly topology with adaptive routing

System Interconnect BW
Peak Injection BW 2.12PB/s
Peak Bisection BW 0.69PB/s

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

Programming Environment
- C/C++, Fortran
- SYCL/DPC++
- OpenMP 5.0
- Kokkos, RAJA

AURORA: SOFTWARE

Three Pillars of Aurora

Simulation	Data	Learning
HPC Languages	Productivity Languages	Productivity Languages
Directives	Big Data Stack	DL Frameworks
Parallel Runtimes	Statistical Libraries	Statistical Libraries
Solver Libraries	Databases	Linear Algebra Libraries

Compilers, Performance Tools, Debuggers
Math Libraries, C++ Standard Library, libc
I/O, Messaging
Containers, Visualization
Scheduler
Linux Kernel, POSIX

Introducing oneAPI Ecosystem

“**oneAPI** is a cross-industry, open, standards-based unified programming model that delivers a common developer experience across accelerator architectures—for faster application performance, more productivity, and greater innovation.”

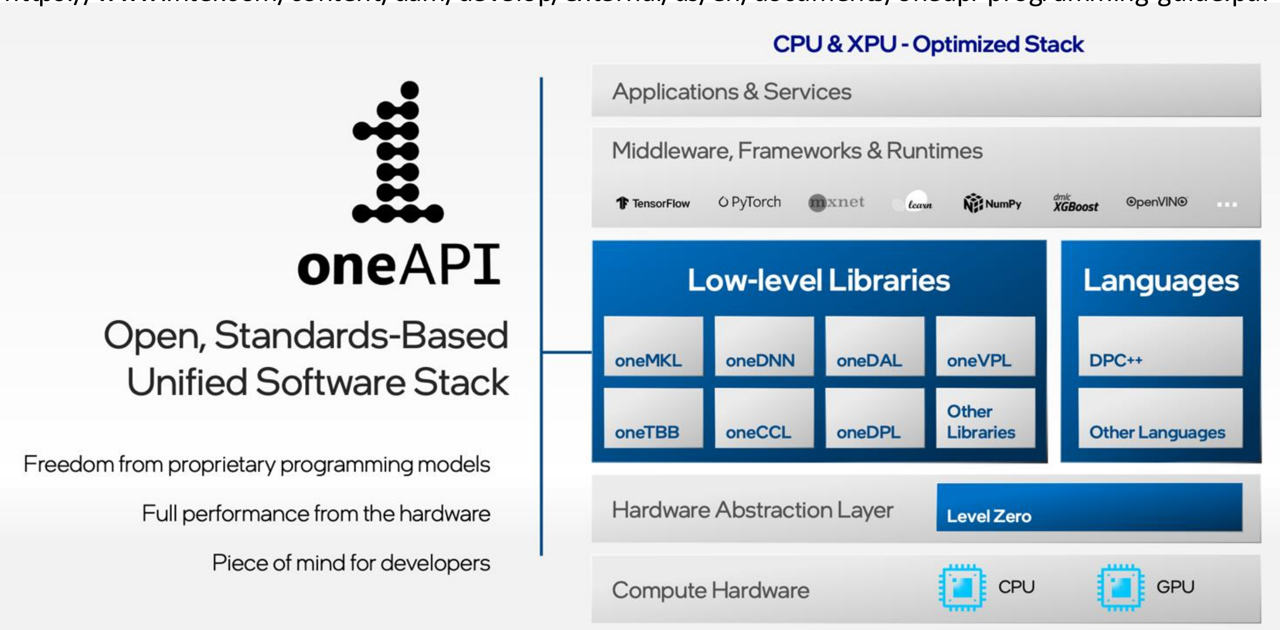
<u>Three Components</u>

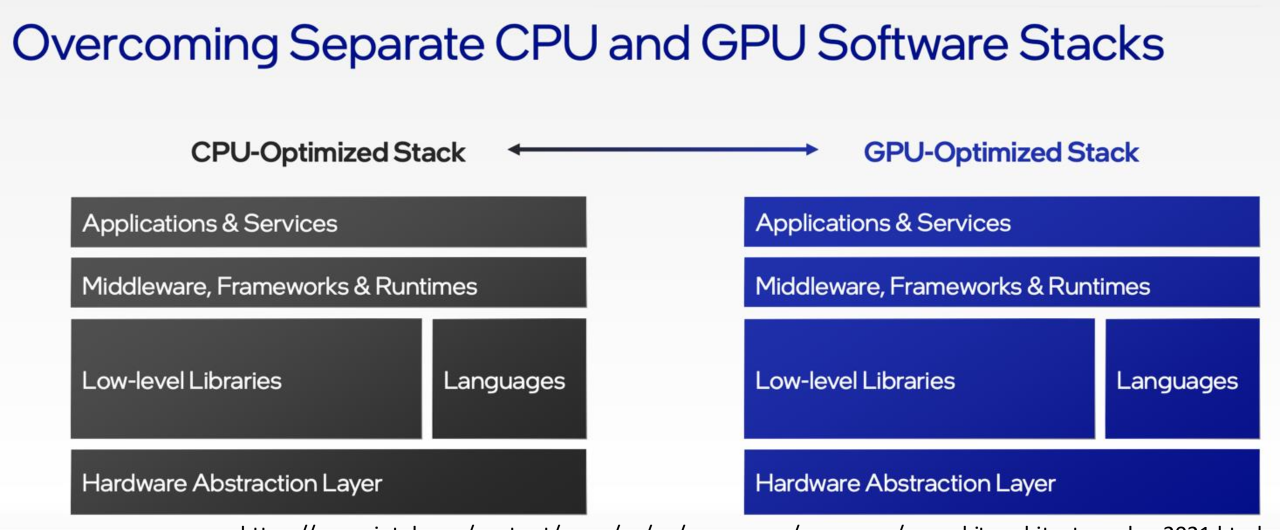
- Language
  - DPC++
- Libraries
  - oneMKL, oneDAL, ...
- Hardware Abstraction Layer
  - Level Zero (L0)

Set of specifications that any one can implement

Intel has their own implementations
https://software.intel.com/ONEAPI

https://www.intel.com/content/dam/develop/external/us/en/documents/oneapi-programming-guide.pdf





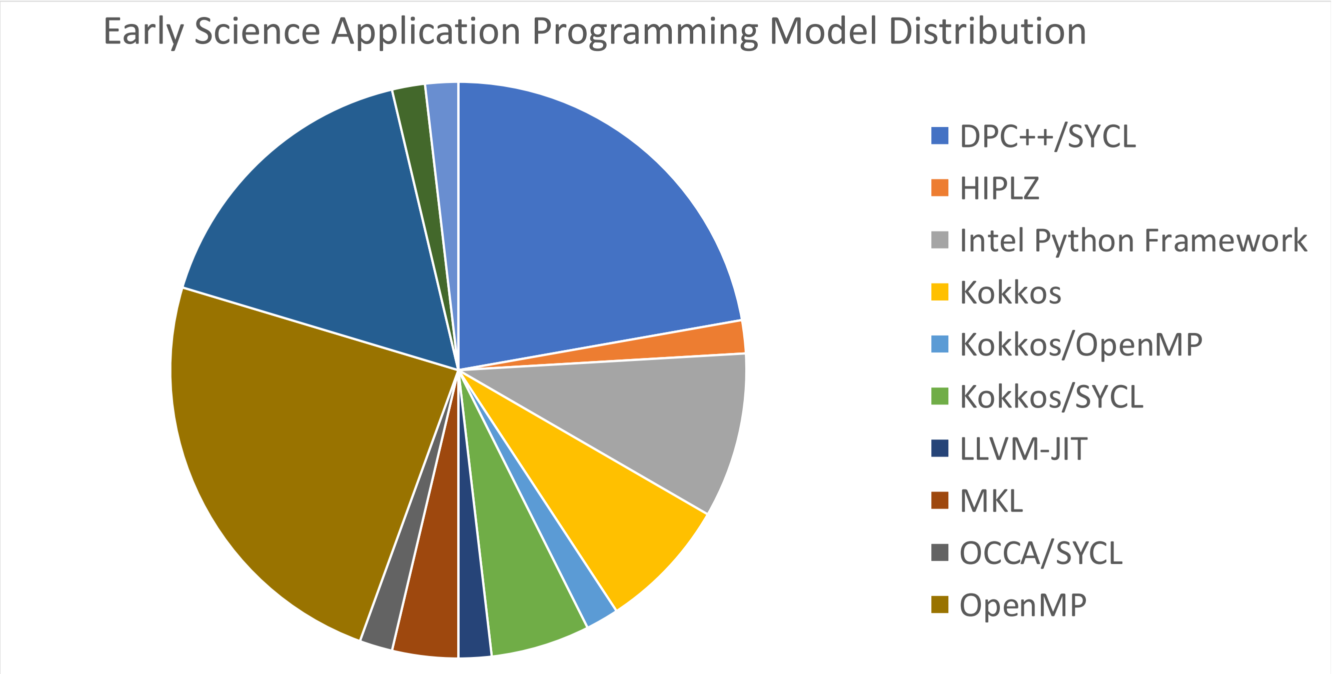
https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html

Aurora Programming Models

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

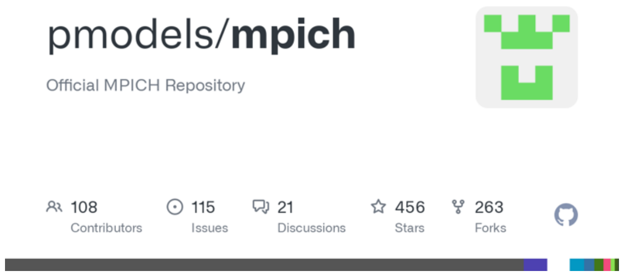
- Experimental
  - HIP

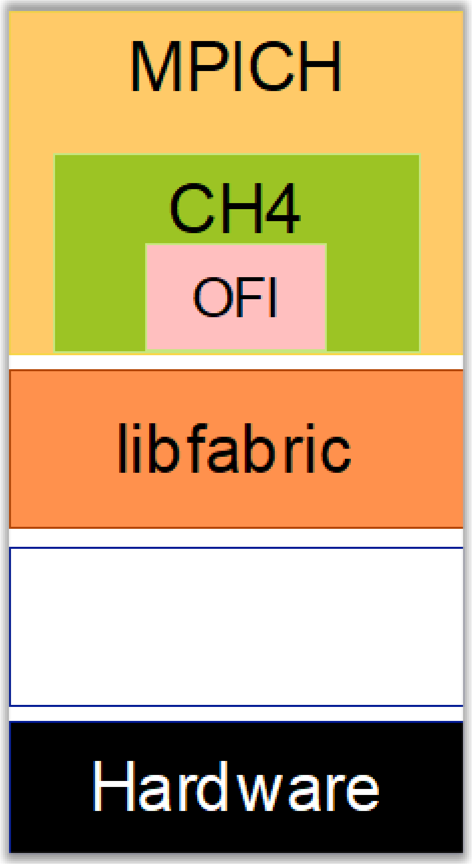
- Not available on Aurora
  - CUDA
  - OpenACC



MPI

- Based on open source MPICH with new features to support Aurora
- Uses OFI (Open Fabrics Interface) to communicate with the Slingshot Interconnect
- Redesigned to reduce instruction counts and remove non-scalable data structures
- Innovative collective algorithms optimized for Dragonfly network topology
- GPU aware for Intel GPUs
  - It is built on top of oneAPI Level Zero
  - It supports point to point, one-sided, and collectives
  - Support for different data types through the Yaksa library
- Intel GPUs and all-to-all connectivity across the GPUs inside the node
- Multiple NICs on the same node
  - Distribution of processes to NICs
  - Striping (a single rank distributes a single message across multiple NICS)
  - Hashing (a single rank sends different messages through different NICs, e.g., depending on the communicator or the target rank)
  - Efficient multithreading support to use multiple NICs





Conclusions

- Challenging design of a Exascale supercomputer
- Intricate system design
- Compute Performance Vs Communication Complexity
- Application scalability balanced by dense compute and hierarchical interconnect

QUESTIONS?

SERVESH@ANL.GOV