Argonne
NATIONAL LABORATORY

An overview of Argonne’s Aurora Exascale Supercomputer and its Programming Models

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

PATH TO EXASCALE

Elements of a supercomputer

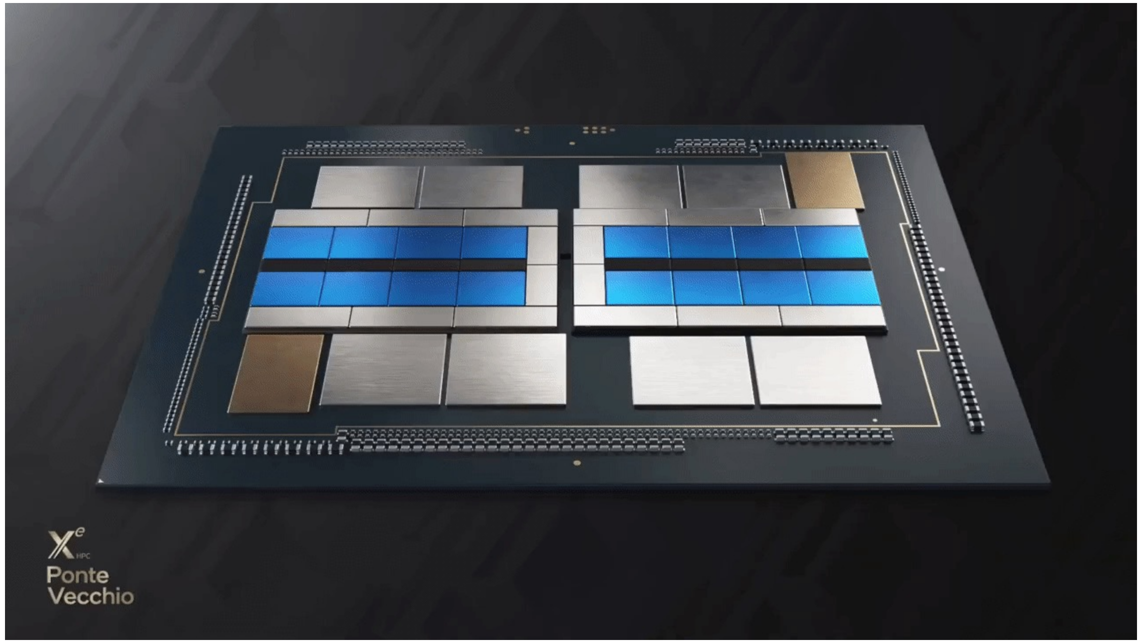
- Processor
  - —architecturally optimized to balance complexity, cost, performance, and power
- Memory
  - —generally commodity DDR, amount limited by cost
- Node
  - —may contain multiple processors, memory, and network interface
- Network
  - —optimized for latency, bandwidth, and cost
- IO System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

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

Argonne NATIONAL LABORATORY

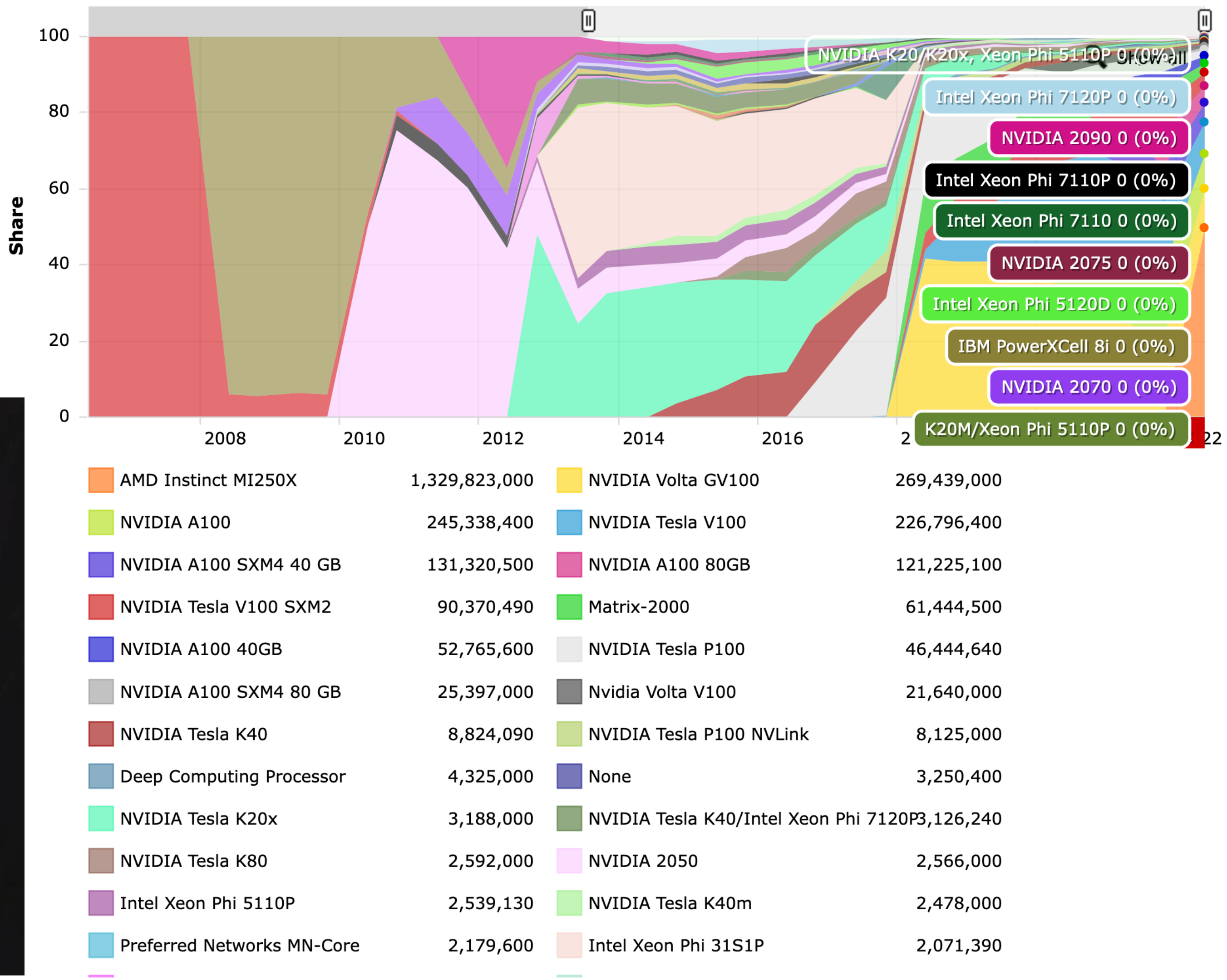
Path to Exascale Computing

- 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

Accelerator/Co-Processor - Performance Share

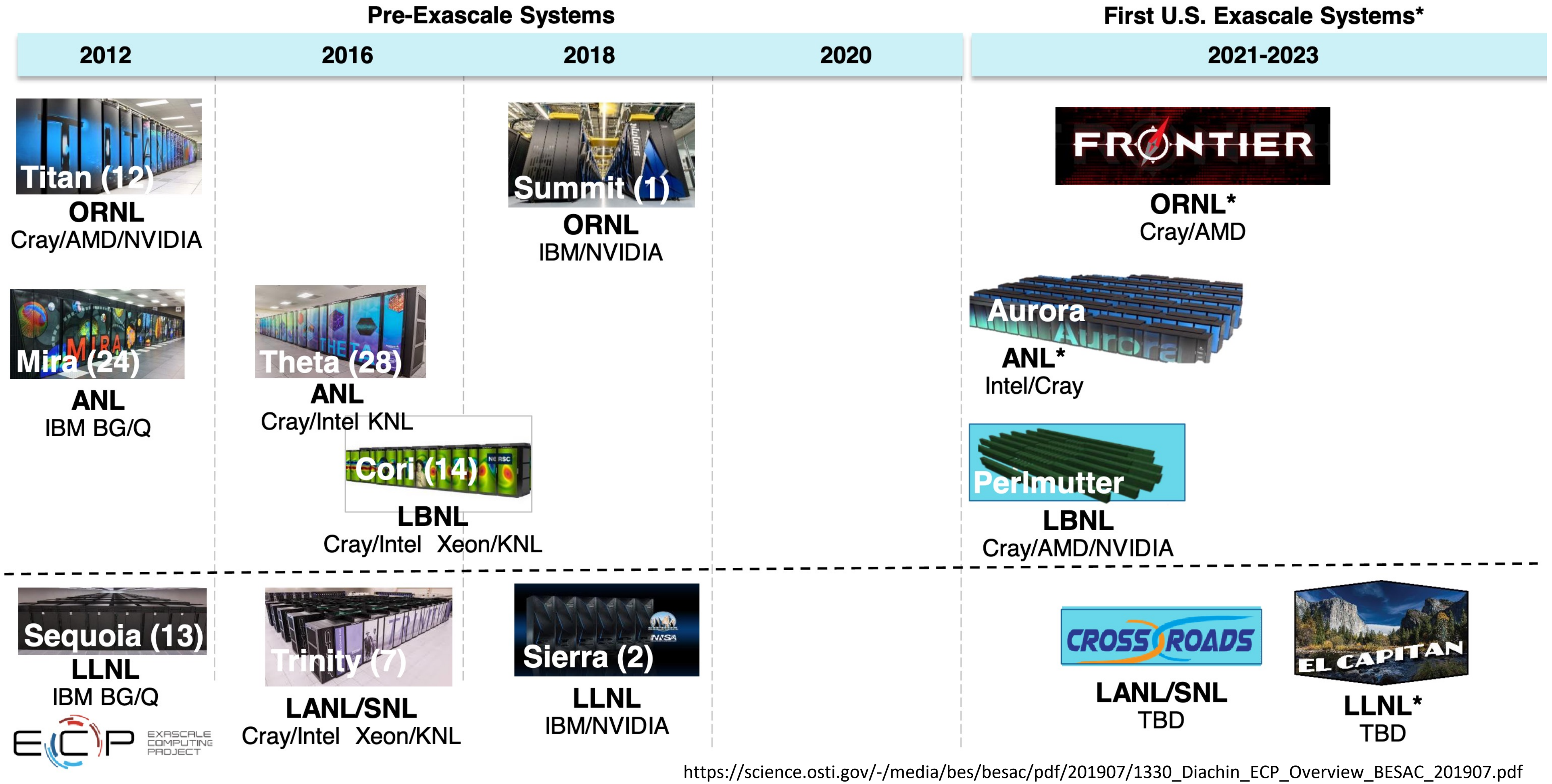


https://www.top500.org/statistics/overtime/

Exascale Computing Project

Department of Energy (DOE) Roadmap to Exascale Systems

An impressive, productive lineup of *accelerated node* systems supporting DOE's mission



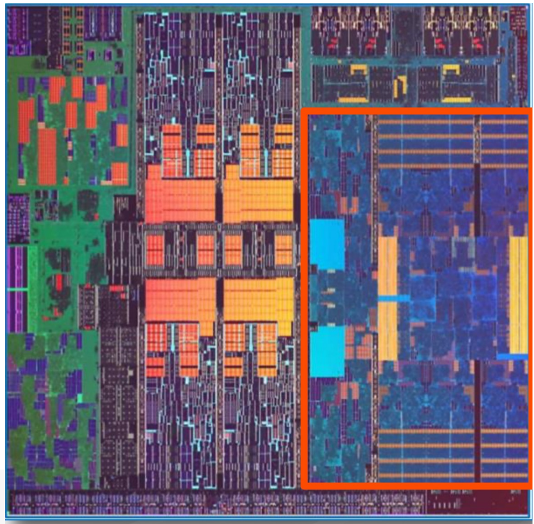
https://science.osti.gov/-/media/bes/besac/pdf/201907/1330_Diachin_ECP_Overview_BESAC_201907.pdf

AURORA: INTRODUCING THE INTEL X^e GPU

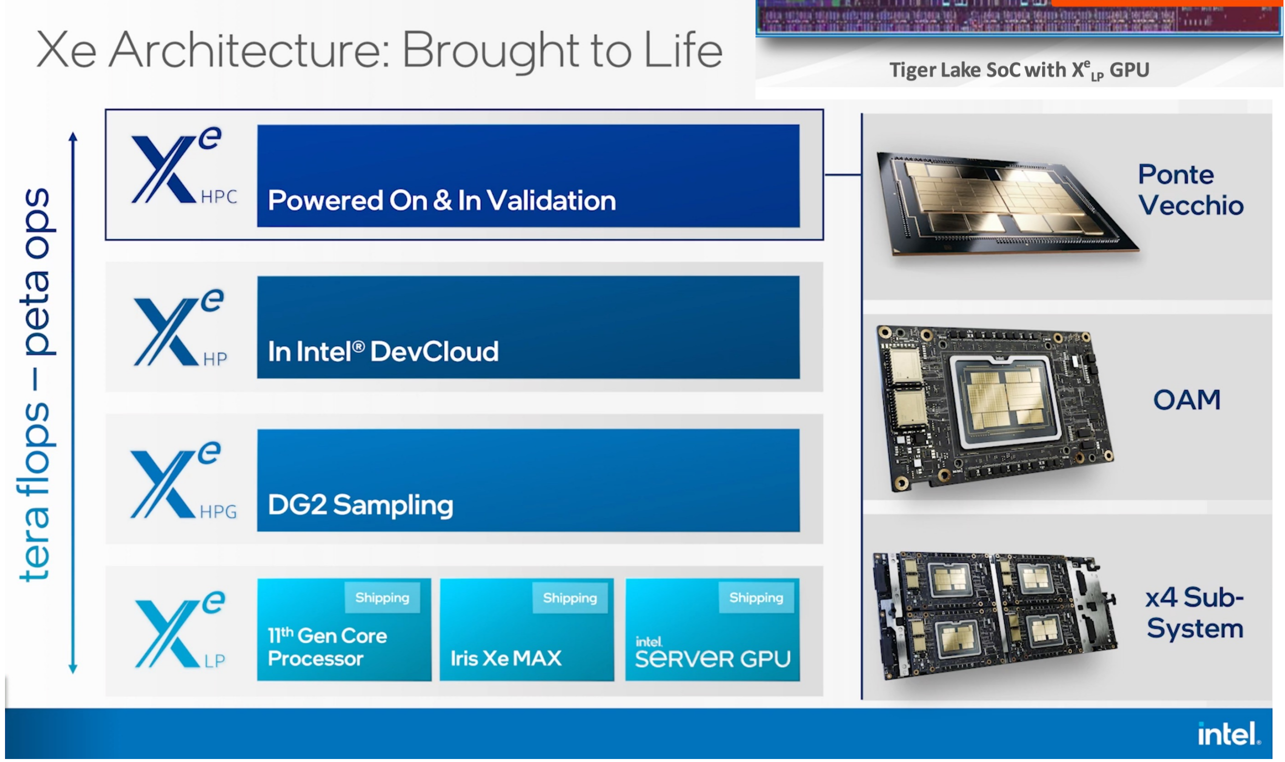
Argonne NATIONAL LABORATORY

Intel GPUs

- ❑ Intel has been building integrated GPUs for over a decade
- ❑ These have evolved into Xe architecture used in next gen GPUs
  - ❑ Xe LP
    - ❑ Platforms: Tiger Lake, Iris Xe Max
    - ❑ Integrated low power
  - ❑ Xe HP/HPG
    - ❑ DG2/Intel Arc GPU
    - ❑ Discrete & High power
  - ❑ Xe HPC
    - ❑ Ponte Vecchio
    - ❑ High performance computing

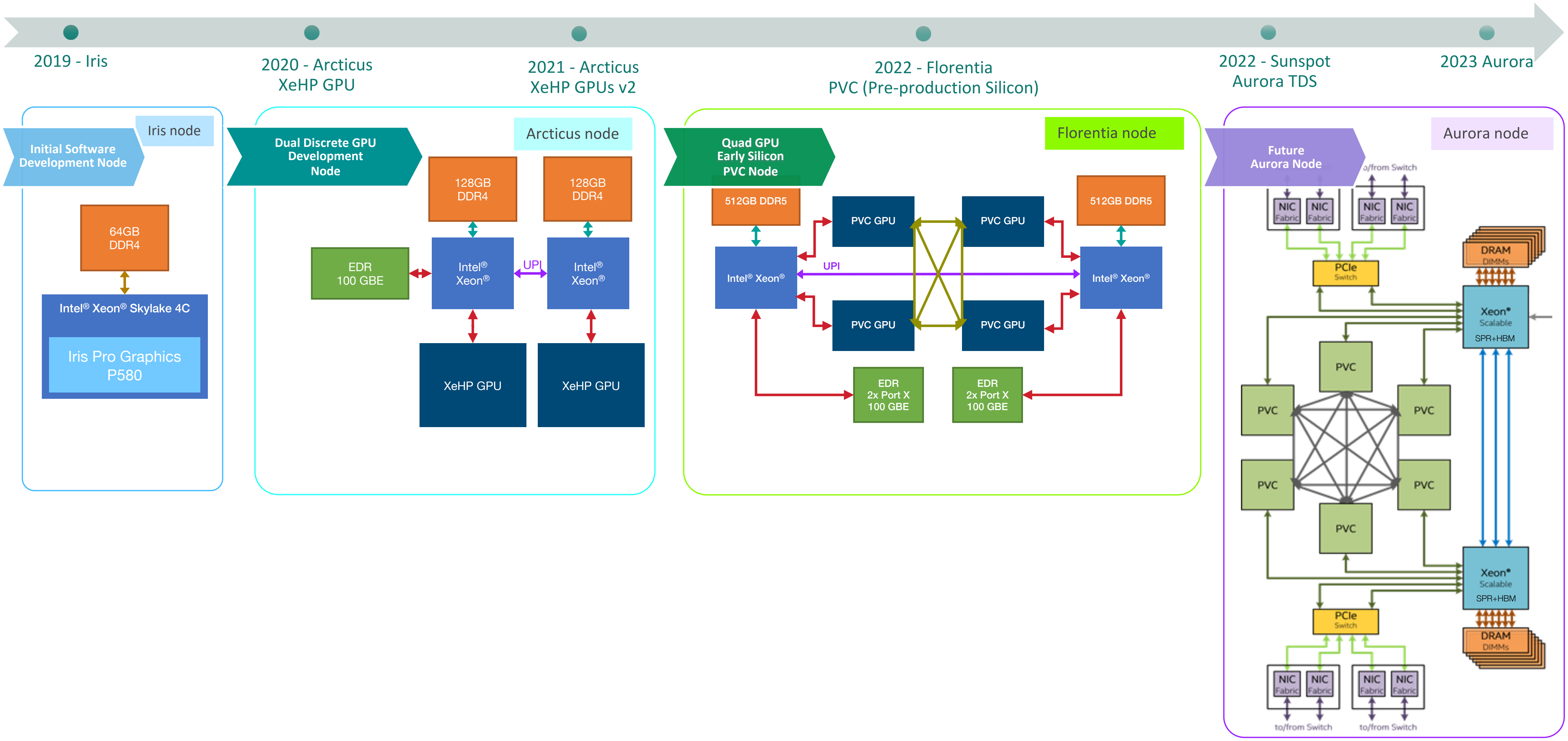


Tiger Lake SoC with X^e_{LP} GPU



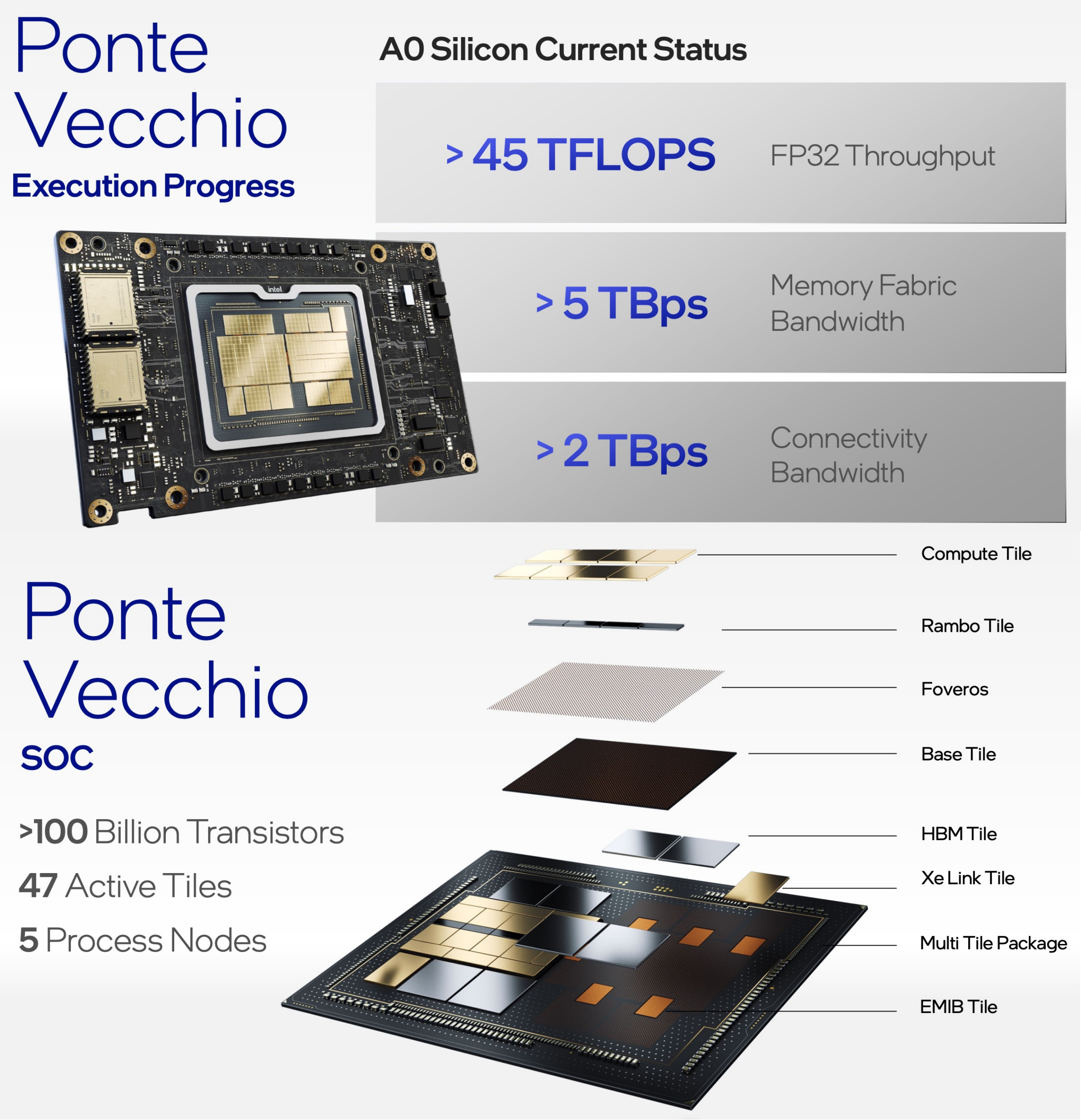
Intel's HPC GM Trish Damkroger Keynotes 2021 ISC
https://www.youtube.com/watch?v=PuEcCRJLrvs

Evolution of Intel GPUs at Argonne



Intel Ponte Vecchio GPU

- ❑ Multi Tile architecture
- ❑ Compute Tile
  - ❑ Xe Cores
  - ❑ L1 Cache
- ❑ Base Tile
  - ❑ PCIe Gen5
  - ❑ HBM2e Main Memory
  - ❑ MDFI
  - ❑ EMIB
- ❑ Connectivity Tile
  - ❑ Xe link

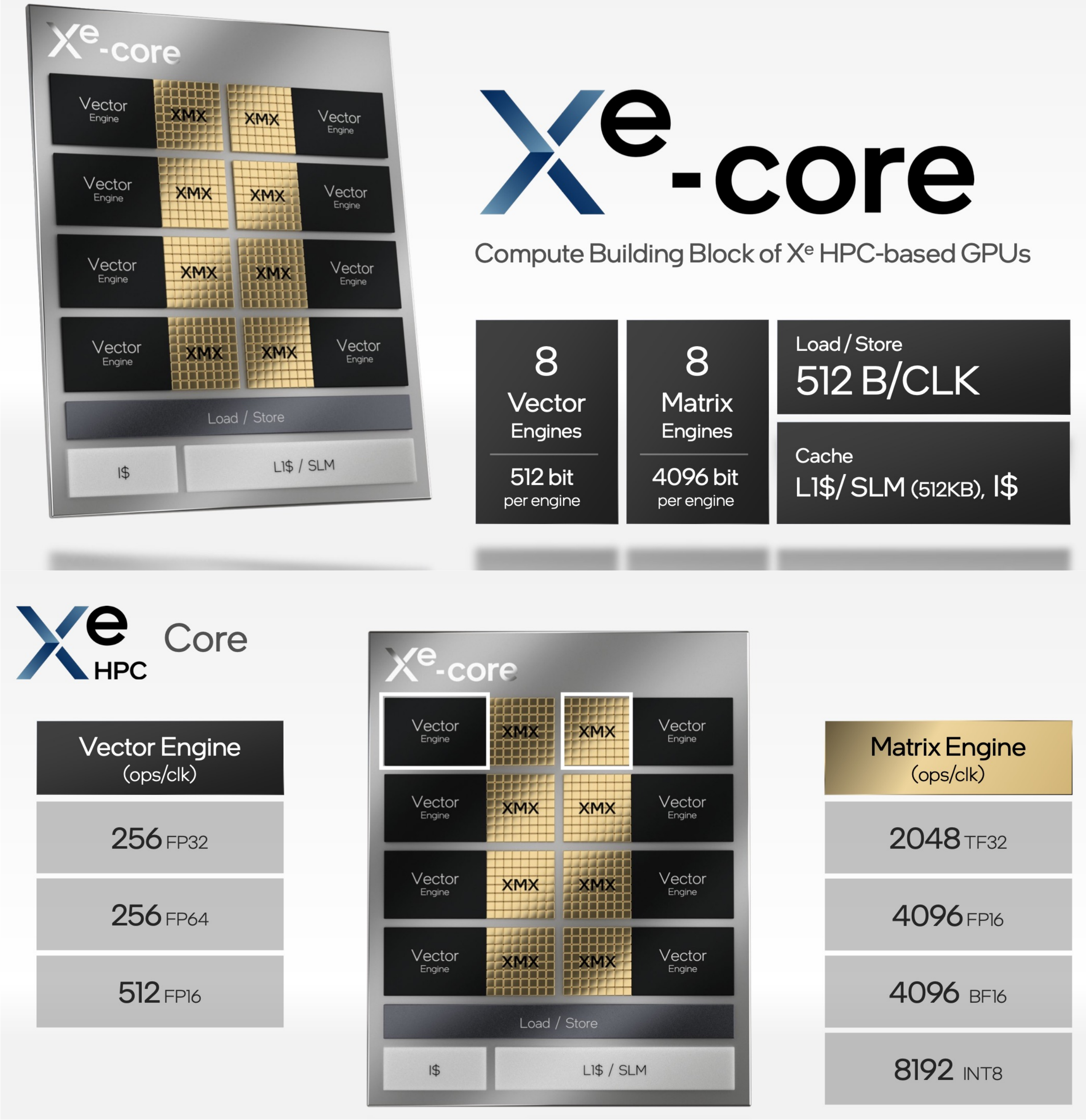


https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html

Xe Core

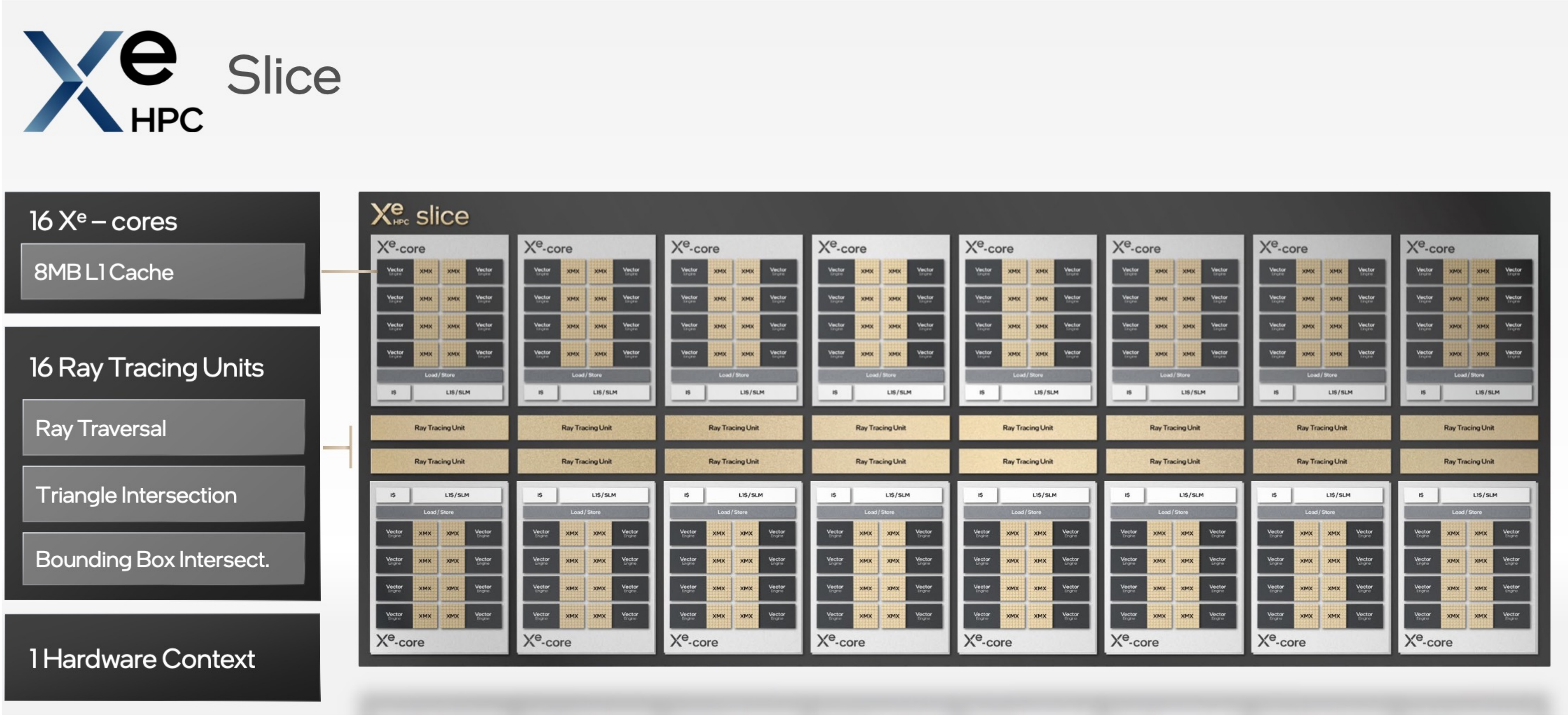
- The fundamental compute block
  - Vector Engine
    - Traditional compute pipeline
  - Matrix Engine
    - Low precision compute pipeline
  - L1 Data Cache
    - Shared Local Memory
  - Instruction Cache

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



Xe HPC Slice

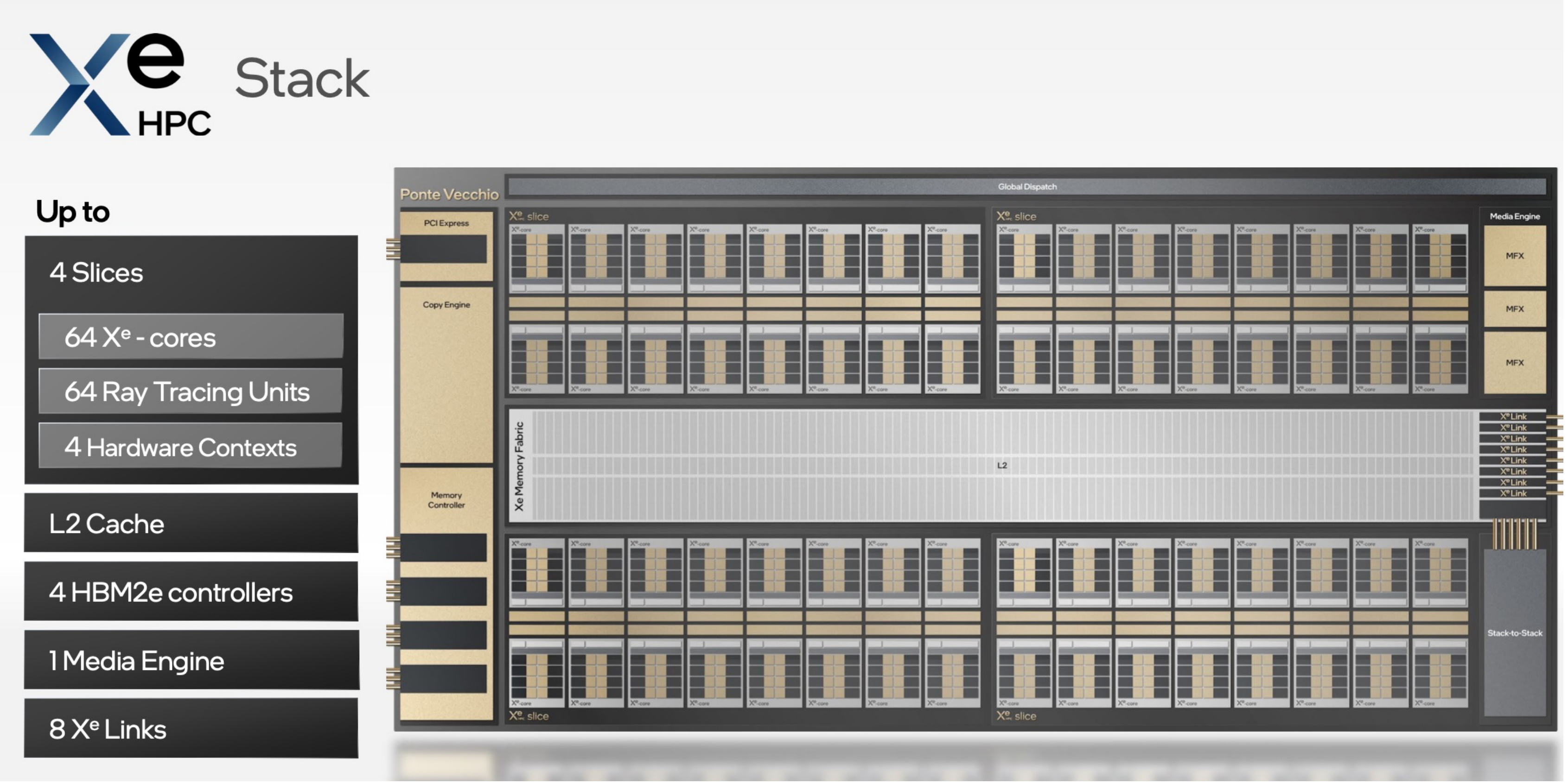
- ❑ Xe Cores
- ❑ Hardware Context
- ❑ Hardware Unit
  - ❑ Ray Tracing



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

Xe HPC Stack

- ❑Multiple Slices
- ❑HBM2e Memory Controller
- ❑Cache Memory Fabric
- ❑PCIe Endpoint
- ❑Hardware specific engines
- ❑Stack to Stack Interconnect
- ❑Xe links
  - ❑Multi GPU Interconnect



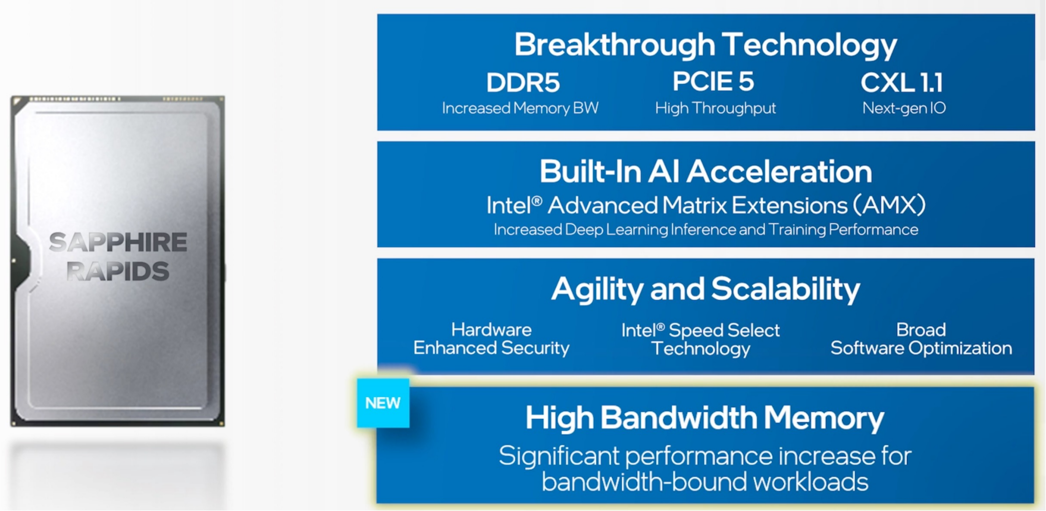
https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html

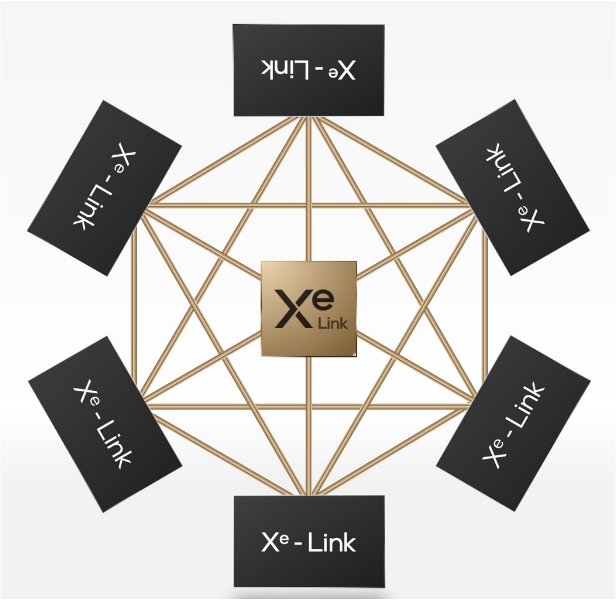
AURORA COMPUTE BLADE

Aurora Compute Blade

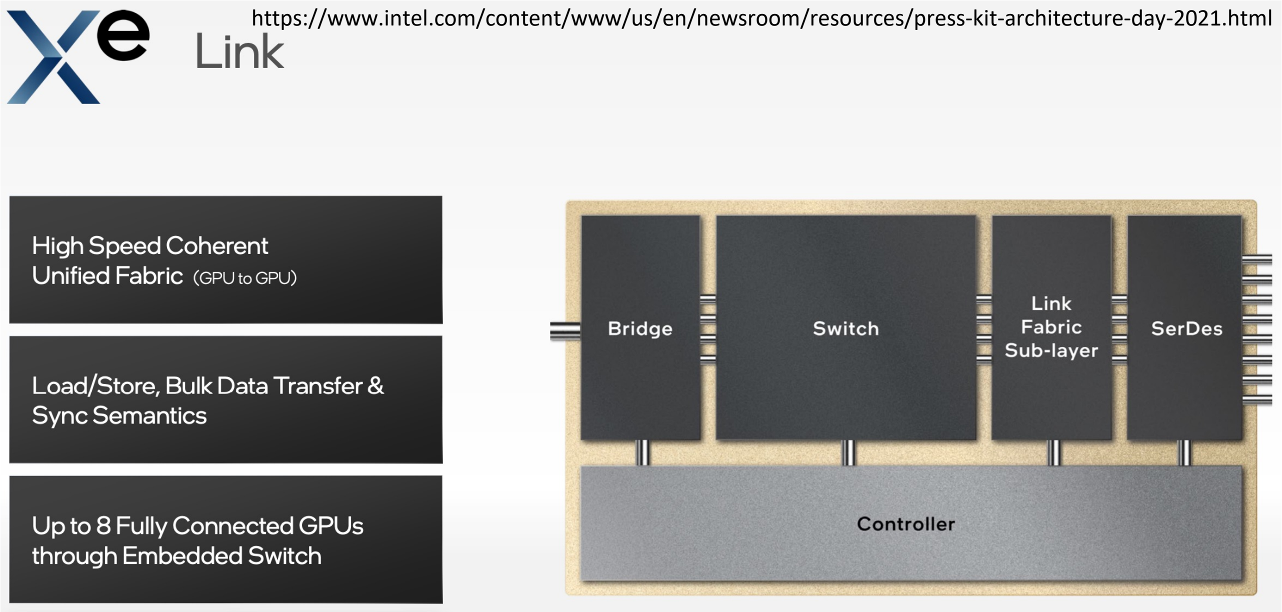
- ❑ Sapphire Rapids CPU
  - ❑ DDR5/HBM
  - ❑ PCIe Gen5
- ❑ GPU – GPU Interconnect
  - ❑ Xe Link

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

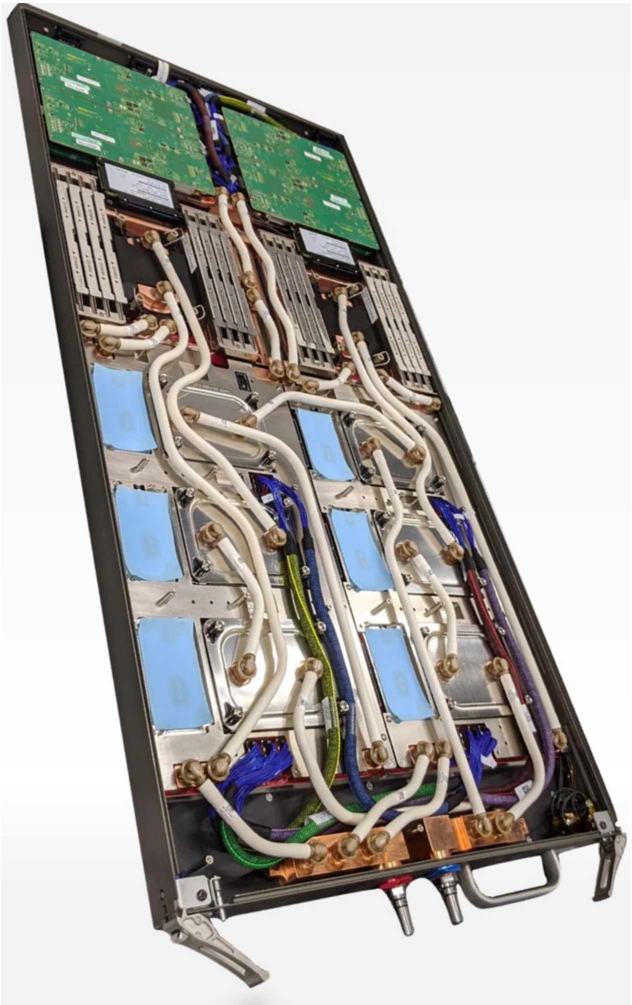


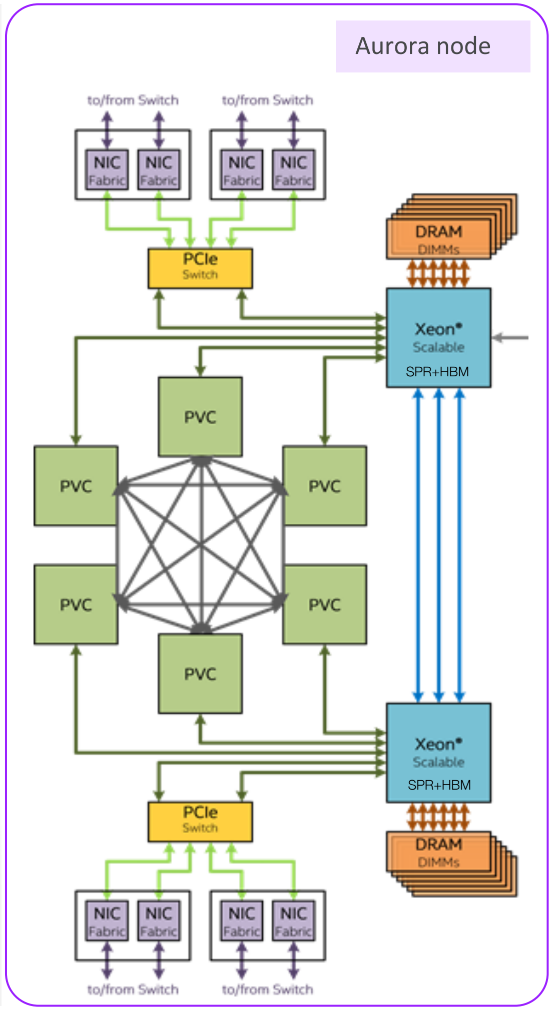


https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html



Aurora Compute Blade





Argonne NATIONAL LABORATORY

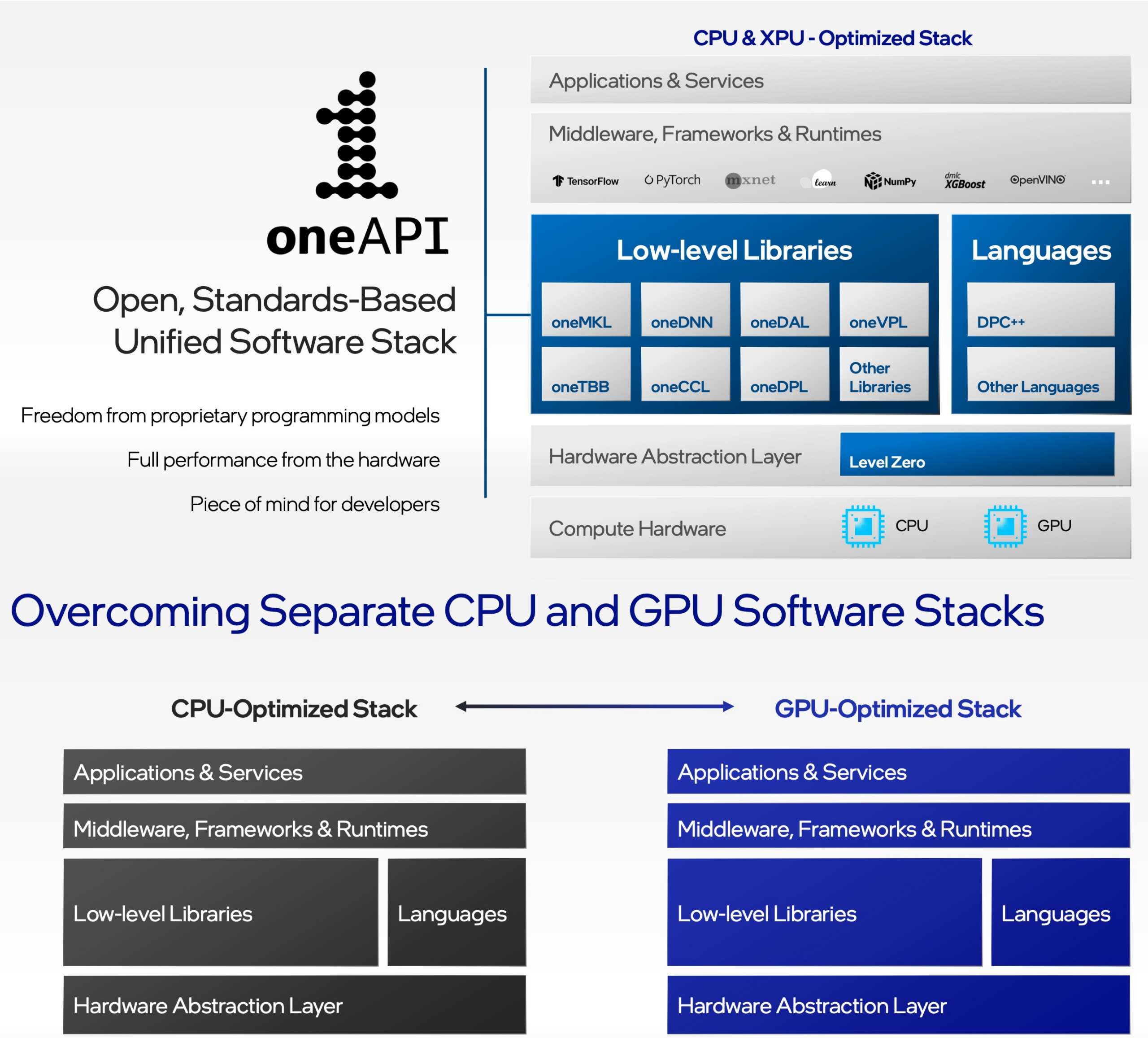
AURORA PROGRAMMING ECOSYSTEM

Argonne
NATIONAL LABORATORY

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

Introducing Intel oneAPI Ecosystem

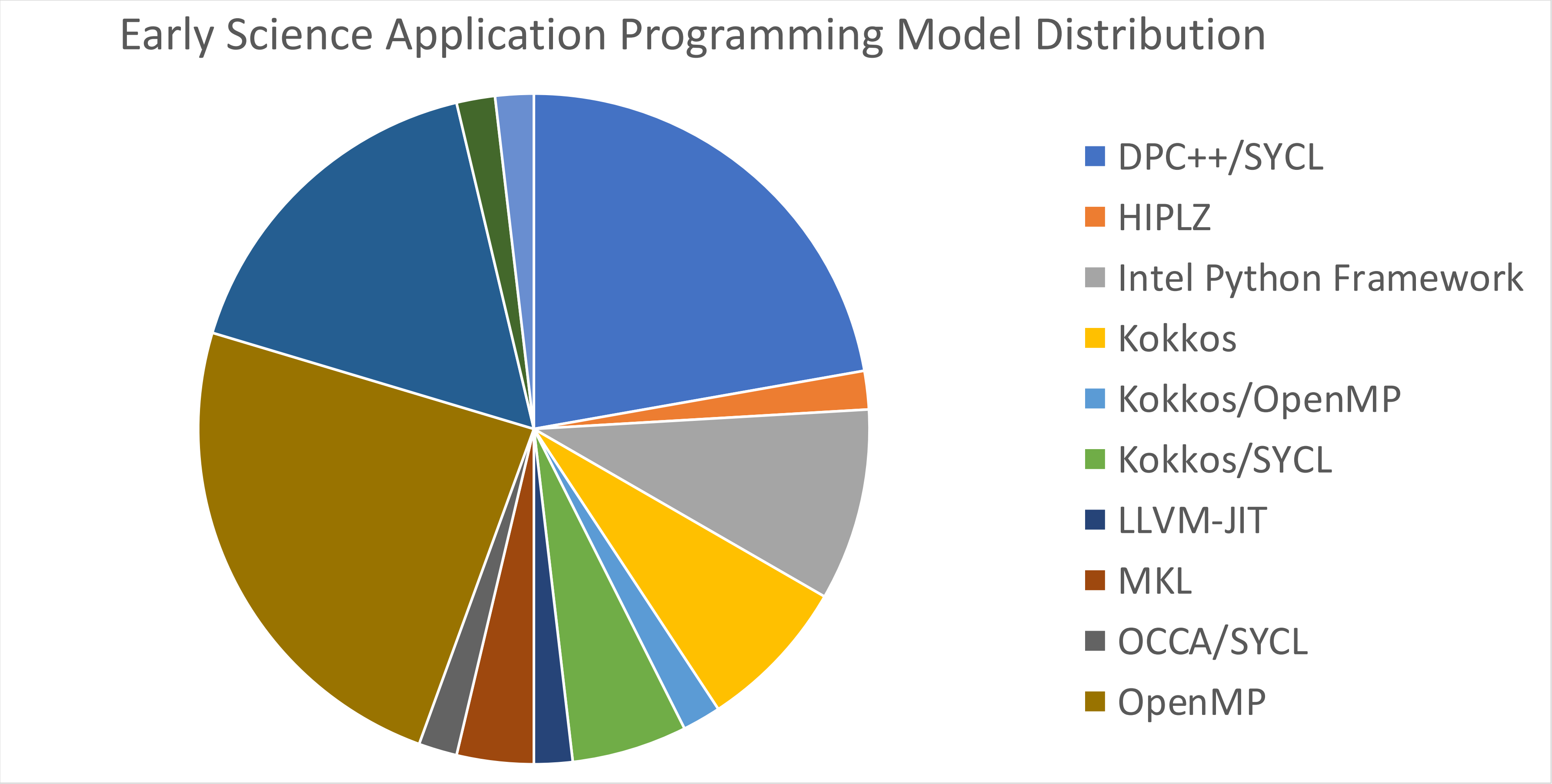
- Unified programming model
- Extensive set of hardware optimized libraries
- New spec for hardware abstraction of accelerators
  - Level Zero
- Many components in the Open Source domain
- AI Framework Ecosystem



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

Programming models

- ❑ Perspective based on early science applications
- ❑ Targeted for Aurora



SYCL/DPC++ Programming Model

- For compilation

```
#SYCL JIT
$icpx –fsycl –std=c++17 -fsycl-targets=spir64 foo.cpp
```

- Code example to get your started
  - —https://github.com/argonne-lcf/sycltrain/tree/master/9_sycl_of_hell
- SYCL Specification
  - —https://www.khronos.org/registry/SYCL/specs/sycl-2020/html/sycl-2020.html
- Intel implementation of SYCL is OpenSource
  - —https://github.com/intel/llvm
- You can mix sycl and openmp

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

sycl::range global{padded_length};
sycl::range local{block_size};

double nstream_time{0};
const size_t bytes = length * sizeof(double);

double * h_A = sycl::malloc_host<double>(length, q);
double * h_B = sycl::malloc_host<double>(length, q);
double * h_C = sycl::malloc_host<double>(length, q);

for (size_t i=0; i<length; ++i) {
  h_A[i] = 0;
  h_B[i] = 2;
  h_C[i] = 2;
}

double * d_A = sycl::malloc_device<double>(length, q);
double * d_B = sycl::malloc_device<double>(length, q);
double * d_C = sycl::malloc_device<double>(length, q);
q.memcpy(d_A, &(h_A[0]), bytes).wait();
q.memcpy(d_B, &(h_B[0]), bytes).wait();
q.memcpy(d_C, &(h_C[0]), bytes).wait();

for (int iter = 0; iter<=iterations; iter++) {
   q.parallel_for(sycl::nd_range{global, local}, [=](sycl::nd_item<1>
it) {
      const size_t i = it.get_global_id(0);
      if (i<length) {
        d_A[i] += d_B[i] + scalar * d_C[i];
      }
   });
   q.wait();
}

q.memcpy(&(h_A[0]), d_A, bytes).wait();
```

https://github.com/jeffhammond/PRK/blob/default/Cxx11/nstream-dpcpp.cc

OpenMP Programming Model

- icpx/ifx is the replacement of icpc/ifort
- OpenMP 5.0+ for C/C++ and Fortran
- Intel documentation: https://www.intel.com/content/www/us/en/develop/documentation/oneapi-dpcpp-cpp-compiler-dev-guide-and-reference/top.html
- Compile:

```
#C/C++ OpenMP JIT
$icpx –fiopenmp -fopenmp-targets=spir64 stream.cpp -o executable

#Fortran OpenMP JIT
$ifx –fiopenmp -fopenmp-targets=spir64 stream.F90 -o executable
```

```
double * RESTRICT A = new double[length];
double * RESTRICT B = new double[length];
double * RESTRICT C = new double[length];

double scalar = 3.0;
#pragma omp target data map(tofrom: A[0:length])\
 map(to: B[0:length], C[0:length]) )
{
  for (int iter = 0; iter<=iterations; iter++) {
    #pragma omp target teams distribute parallel for simd
    for (size_t i=0; i<length; i++) {
      A[i] += B[i] + scalar * C[i];
    }
  }
}
```

https://github.com/jeffhammond/PRK/blob/default/Cxx11/nstream-openmp-target.cc

Kokkos Programming Model

- Kokkos C++ Performance Portability Programming EcoSystem
- https://github.com/kokkos/kokkos
- Abstractions for both parallel execution of code and data management
- Intel GPU and CPU support via OneAPI SDK

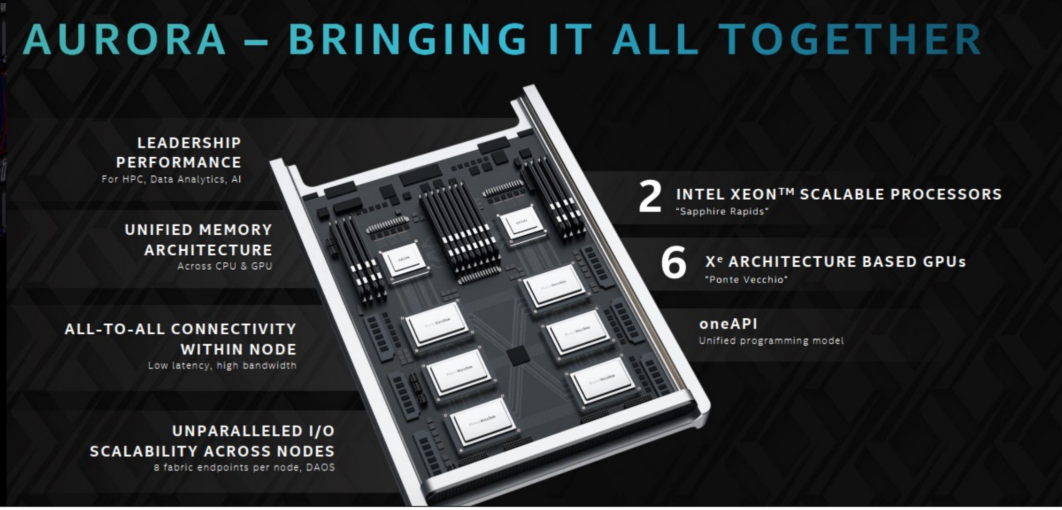
```
vector A("A", length);
vector B("B", length);
vector C("C", length);

const double scalar(3);
{
  Kokkos::parallel_for(length, KOKKOS_LAMBDA(size_t const i) {
      A[i] = 0.0;
      B[i] = 2.0;
      C[i] = 2.0;
  });
  Kokkos::fence();

  for (int iter = 0; iter<=iterations; ++iter) {
    if (iter==1) {
      Kokkos::fence();
    }
    Kokkos::parallel_for(length, KOKKOS_LAMBDA(size_t const i) {
        A[i] += B[i] + scalar * C[i];
    });
  }
  Kokkos::fence();
}
```

https://github.com/jeffhammond/PRK/blob/default/Cxx11/nstream-kokkos.cc

AURORA SYSTEM OVERVIEW



Peak Performance
≧ 2 Exaflops DP

Intel GPU
Ponte Vecchio (PVC)

Intel Xeon Processor
Sapphire Rapids with High Bandwidth Memory

Platform
HPE Cray-Ex

Compute Node
2 Xeon SPR+HBM processors
6 Ponte Vecchio GPUs
Node Unified Memory Architecture
8 fabric endpoints

GPU Architecture
Intel XeHPC 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

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

Aurora Cabinets Installation at Argonne

Argonne NATIONAL LABORATORY

QUESTIONS?