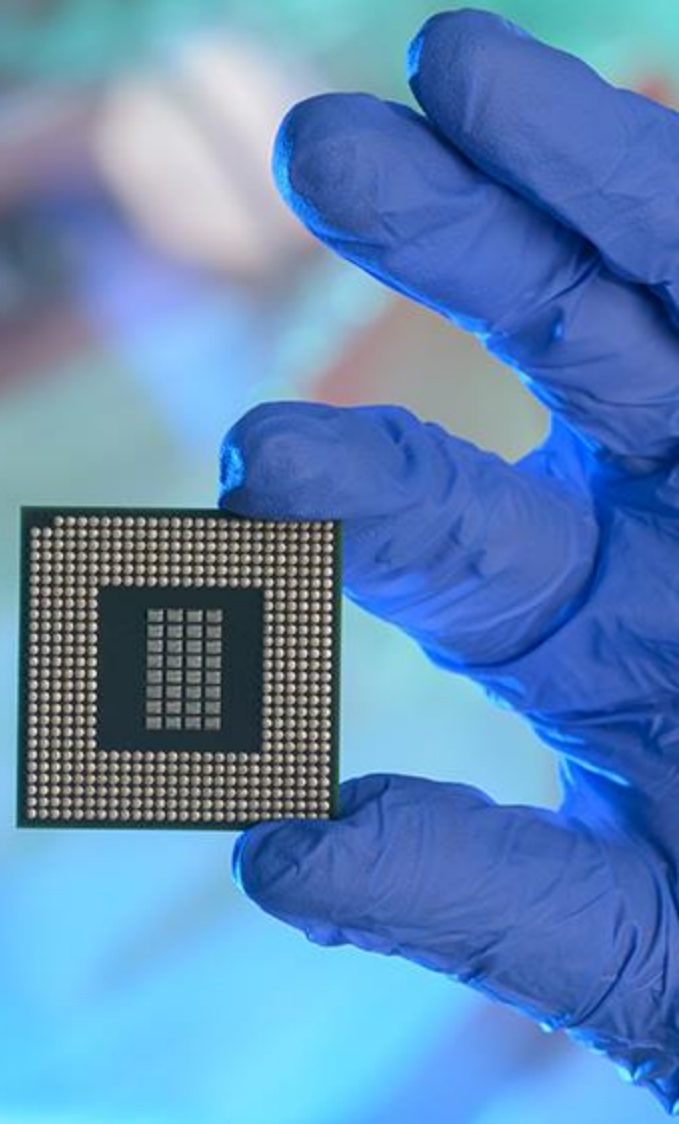
IXPUG Annual Conference 2025

The Evolution of Intel Developer Software

Sanjiv Shah

Vice President, Data Center and AI Group
General Manager, Developer Software Engineering



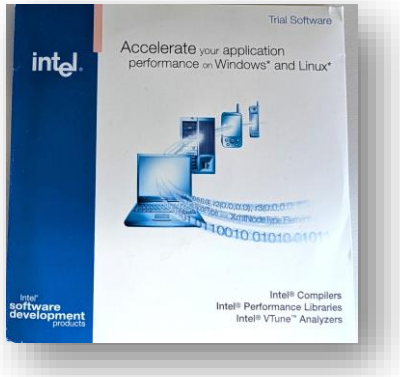
intel

1980s

First Fortran
& C compilers | Debugger

1990s

C++ compiler |
Math Kernel Library (MKL) |
Integrated Performance Primitives (IPP) |
VTune



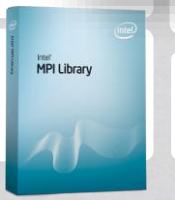
2000 - 2005

OpenMP | Kuck and Associates acquisition

Acquired Compaq Fortran compiler

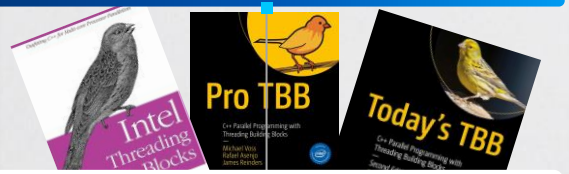
Thread Checker/Inspector | Threading tools

Cluster Tools | Pallas acquisition



Intel® MPI Library

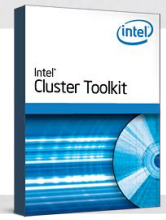
2006 - 2010



Threading Building Blocks

Graphics Performance Analyzers

First Toolkit introduced



Cilk Arts & RapidMind acquisitions | Array Building Blocks

Launch of Intel® Xeon Phi ™

2011 - 2015

Data Analytics Library

Python optimizations

2016 - 2019

Deep Neural Network Library

Open Fabrics Interface | libfabric

Collective Communications Library

2020 - 2025

oneAPI launch | SYCL* chosen as programming model

Codeplay acquisition | oneAPI for Nvidia and AMD GPUs plug-ins

SYCL Graph and joint matrix features

Intel® SHMEM

Our Recent Focus: Improving our Foundational Technology

Compilers

- Shift to LLVM*
- Improved performance
- Added sanitizers

Programming Models

- OpenMP*
  - Offload support
- Threading Building Blocks
  - C++11 rewrite
- Introducing SYCL*
- SYCL* interfaces for libraries

Distributed Programming

- Intel® MPI Library
  - GPU initiated comm
- Fabrics
- oneAPI Collective Communication Library (oneCCL)
- Intel® SHMEM

Analyzers

- Intel® VTune Profiler
  - CPU → whole system analysis
  - Client/server model
- Intel® Advisor
- Intel® Inspector
  - Shift to Sanitizers

LLVM Powering the Next Generation of Compilers

Faster Compile Times

- Fortran: Up to 18% faster over ifort
- C/C++: Up to 16% faster over icc

Improved Diagnostics

- Easier-to-understand C++ error messages
- Enhancements to optimization reports
- Sanitizers

Key Optimization

- Leverage LLVM Optimizations
- Tuned Vectorization & Loop Transformations

Accelerator Support

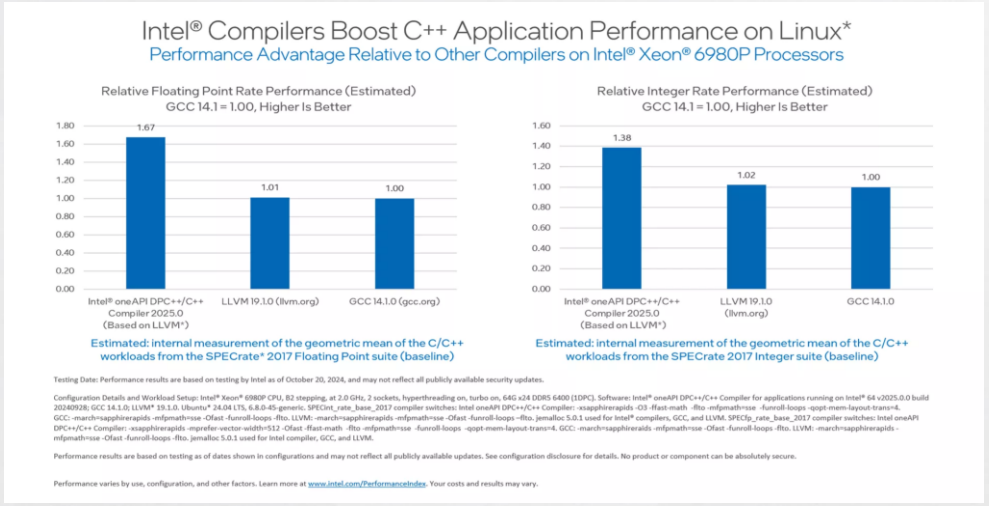
- OpenMP offload for GPUs
- SYCL for CPU & GPU

Openness

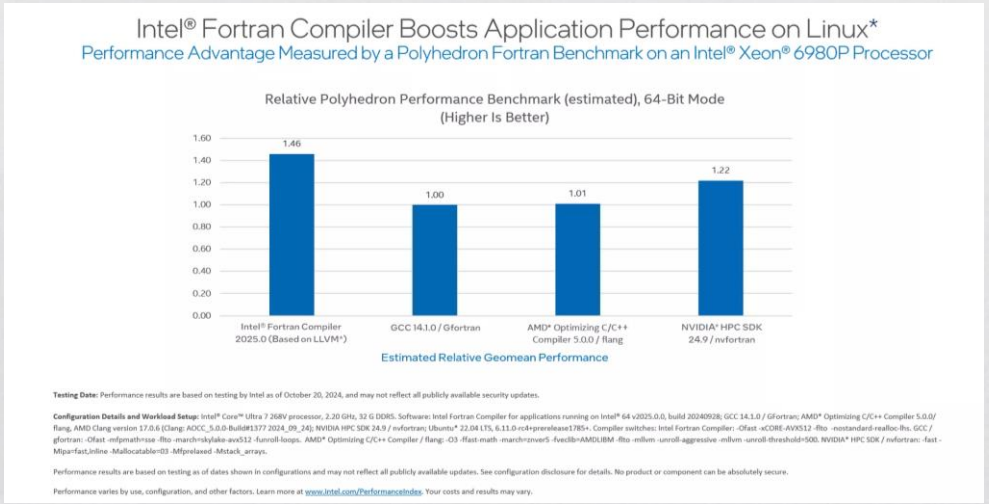
- Languages & Open Standards
- Community engagement & contributions
- Industry adoption
- Friendly licensing

Intel® oneAPI DPC++/C++ and Fortran Compilers

- Intel LLVM-based compilers deliver better performance vs competitive platforms
- Better language integration: C, C++, Fortran, MLIR, OpenCL, Rust, Julia, Java (Tornado JVM)
- Greater collaboration in vectorizer and loop transformation optimizations



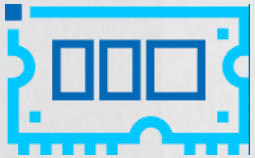
Up to 60% improvement for C/C++

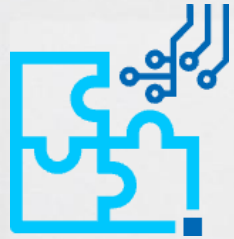
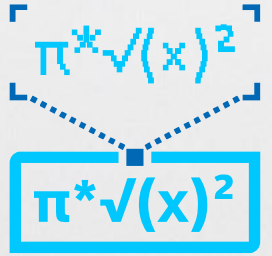


Up to 40% improvement for Fortran

LLVM Sanitizers

Empowering Developers to Build Robust Applications

Address Sanitizer (ASan)	Memory Sanitizer (MSan)	Leak Sanitizer (LSan)	Undefined Behavior (UBSan)	Thread Sanitizer (TSan) + Archer for OpenMP	Numerical Sanitizer (NSan)
Find and fix memory-related errors	Identify uninitialized memory	Find and fix memory leaks early	Detecting undefined behavior in executed code	Identifying data races and synchronization issues	Find and fix floating point computation issues

OpenMP®

Check out the *OpenMP in oneAPI* talk by Jeongnim Kim & Ye Luo @ 12:00pm CDT

Open Source and Community-Driven

- Multi-platform shared memory programming model
- Portable from supercomputer to laptop
- Liboffload community collaboration

Latest Specifications

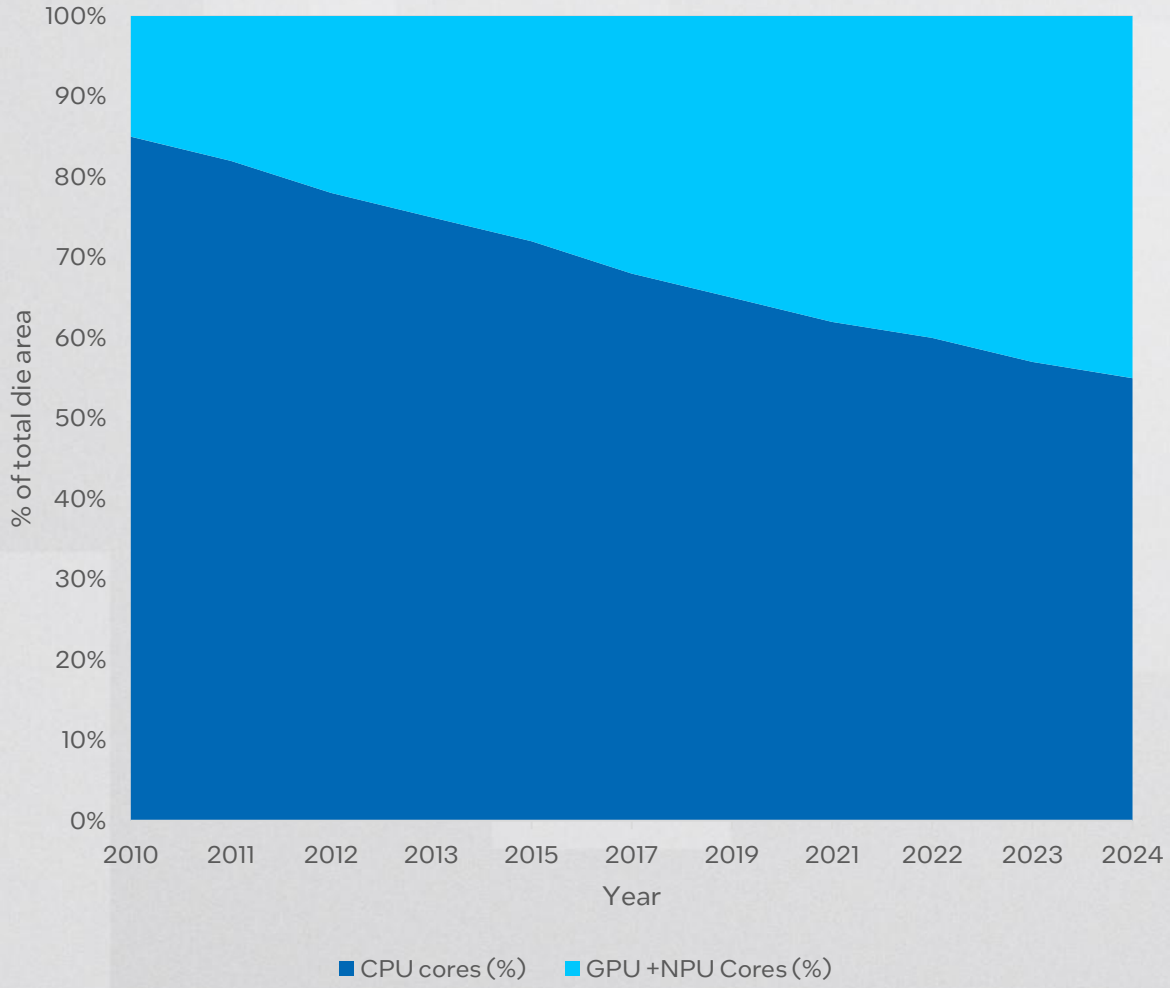
- Version 5.x — Fully supported
  - Full support for accelerator devices
  - Tools and debugging
  - Memory management and tasking
- Version 6.0 — Launched Nov'24. 30% support.
  - Enhanced device support
  - Improved loop transformations
  - Greater control of storage resources and memory spaces
  - Interoperability with SYCL/DPC++

Foundation for AI on CPU

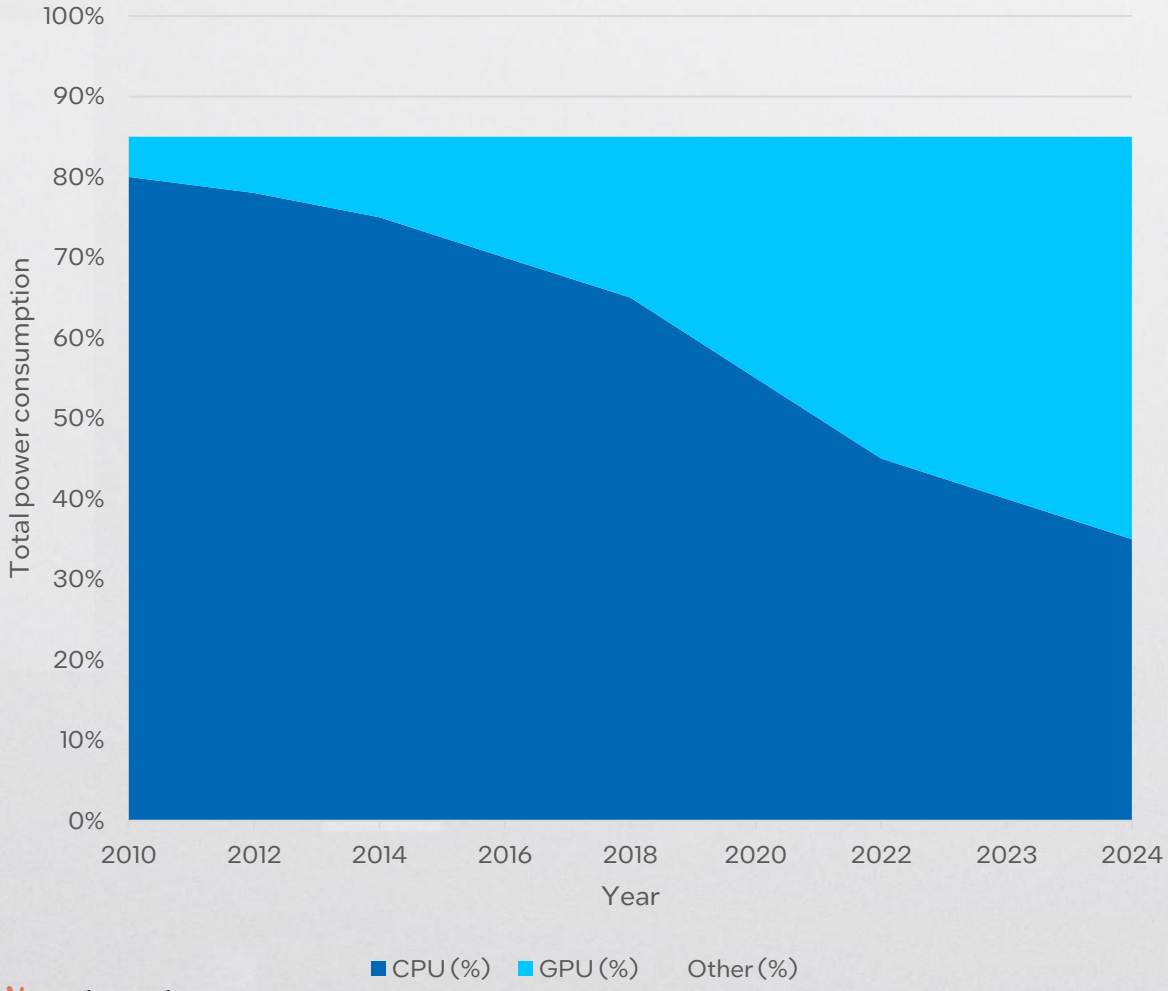
- PyTorch for CPU is built on OpenMP
  - Inductor backend
  - ATen operators

Growth of Accelerators

Intel Client CPU vs GPU+NPU Die Area Allocation (2010-2024)



Server Power Consumption Distribution (2010-2024)



Generated using Claude

Claude.ai (April 14, 2025). "On intel client cpu, plot over time the area of CPU cores versus GPU or NPU cores and map as percentage of total" [AI-generated table of values]
Claude.ai (April 14, 2025). "in a modern server, show me a plot over time of how much power is consumed by GPUs versus CPUs and map it as percentage of total" [AI-generated table of values]
Anthropic: https://claude.ai/chat/b4283459-bc58-40bf-8f84-fc5df62cc148

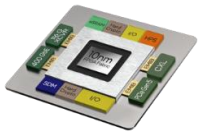
The accelerator programming challenge

Accelerator programming for compute is proprietary and locks you in

CPU

GPU

FPGA



Other Accelerators

48% of developers target systems that use more than one kind of processor or core[1]

Developer Challenges: Multiple Architectures, Vendors, and Programming Models

- You don't have freedom to integrate different technologies together.
- You can't choose the best architecture for your workload.
- It's much harder to bring new innovations in performance to market.

Source 1: Evans Data Global Development Survey Report 23.2 2023

An Open Software Platform

Unify the compute accelerator ecosystem around open standards

2014 | 2019 | 2020 | 2021 | 2022 | 2023 | 2025

SYCL™

Language for open portability

oneAPI™

Build out an open ecosystem at scale

- oneDPL — Data Parallel C++ Library
- oneDNN — Deep Neural Network Library
- oneCCL — Collective Communications Library
- oneDAL — Data Analytics Library
- oneTBB — Threading Building Blocks
- oneMath — Math Kernel Library

UXL — Unified Acceleration Foundation

Open industry governance

Unified Acceleration (UXL) Foundation

Mission: Unify the compute accelerator ecosystem around open standards

- Governance: Linux Foundation's Joint Development Foundation
- Goal: broad-based industry participation and contributions

Steering Members

arm BROADCOM FUJITSU GE HealthCare intel
Imagination Qualcomm SAMSUNG Google Cloud

General Members

सी डैक CDAC micron SK hynix

Contributing Members

25+ contributing members
9 new in past 6 months

Governance & Structure

Courtesy of UXL Foundation: uxlfoundation.org

UXL
Unified Acceleration Foundation

open, multiarchitecture, multivendor programming

- Open industry specification
- Freedom in hardware choice
- Performance, productivity & portability
- Standards-based, community-driven

oneAPI

Intel's implementation with a set of tools

- Optimized for Intel hardware
- Proven performance, best-in-class capabilities
- Supports Fortran, Python, OpenMP, MPI...
- Download free, commercial support available

Implementing SYCL™

- Open Standard from Khronos Group
- Started in 2014, major change in 2020
- Driven by multiple vendors and users
- Single programming model for CPUs, GPUs, and other accelerators
- Enable HW-specific performance
- Interop with other programming models
- Intel's implementation in DPC++/C++ compiler
- Conformant with SYCL 2020

Based on modern C++

Unified address space

```
sycl::malloc_host
sycl::malloc_device
sycl::malloc_shared
```

Atomic ops on non-atomic objects

```
sycl::atomic_ref
```

Subgroup and work group algorithms

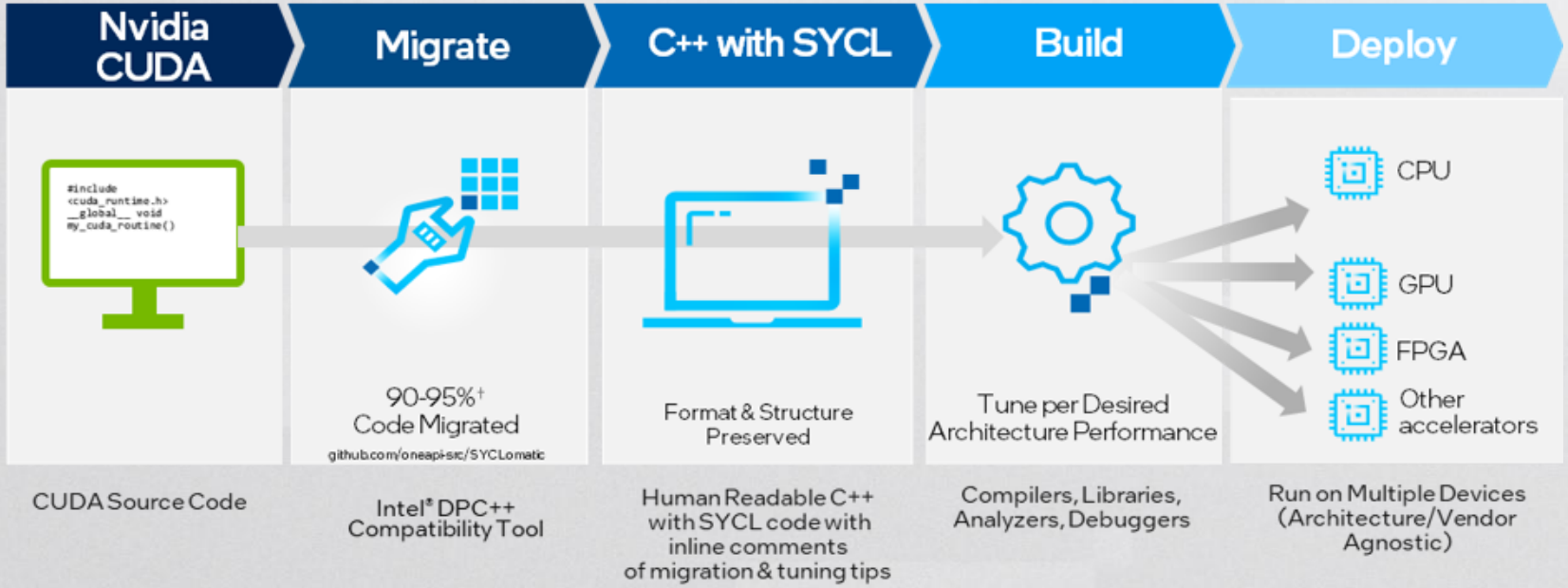
```
Any_of, all_of, non_of, reduce, exclusive_scan,
inclusive_scan, shift_left, shift_right, select, permute
```

Reductions

```
sycl::reduction
```

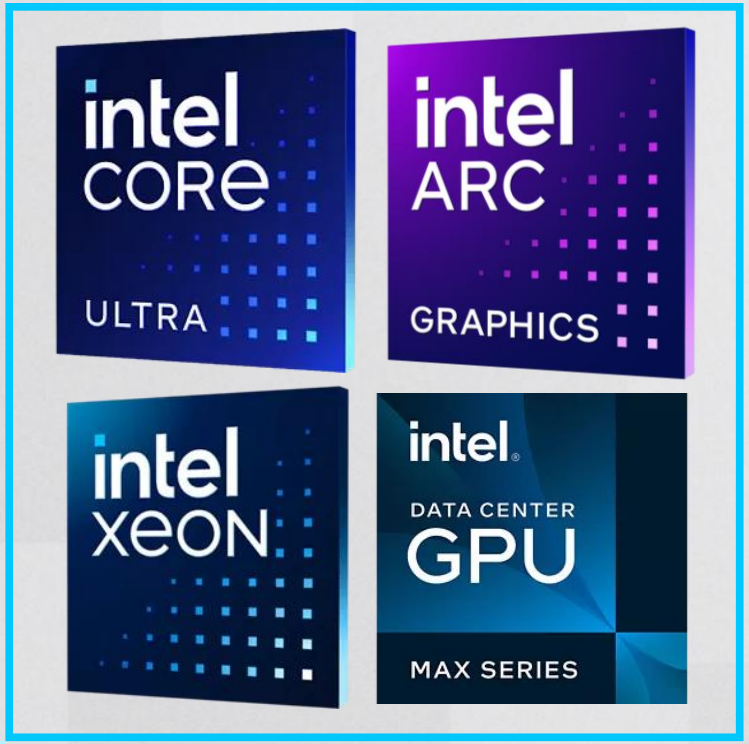
Creating C++ SYCL applications

- Write from scratch or port from CUDA*
- Single codebase runs on accelerators from multiple vendors
- SYCLomatic aids in migrating code from CUDA applications



†Intel estimates as of March 2023. Based on measurements on a set of 85 HPC benchmarks and samples, with examples like Rodinia, SHOC, PENNANT. Results may vary.
*Other names and brands may be claimed as the property of others. SYCL is a trademark of the Khronos Group Inc.

Run SYCL applications everywhere



Intel CPUs and GPUs

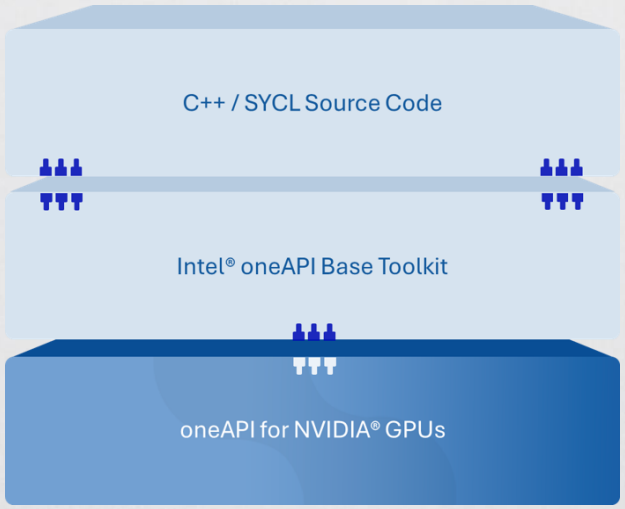
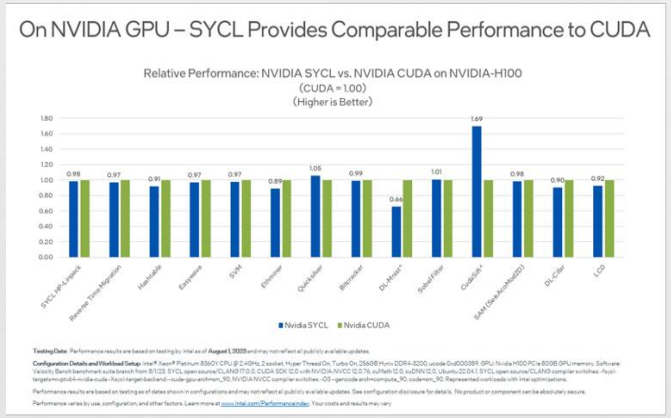


Image courtesy of Codeplay Software Ltd.

Nvidia* GPUs via Codeplay binary plugins to Intel® oneAPI DPC++/C++ Compiler



Comparable performance to CUDA*

Applications and Frameworks using SYCL today

SYCL everywhere: Continuity from supercomputers to laptops

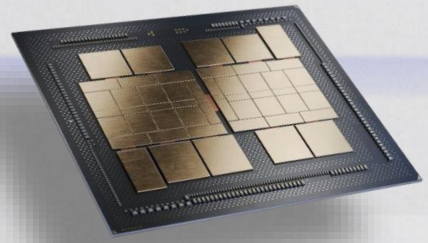
Amber Molecular Dynamics

FAST. FLEXIBLE. FREE. GROMACS

SeisSol

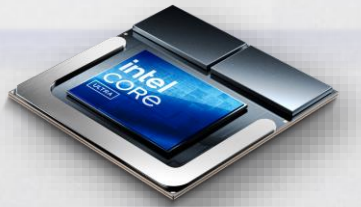
Ansys | Fluent*

Commercial Rendering Application



Intel® Data Center GPU Max

Intel® Arc™ B-series Graphics

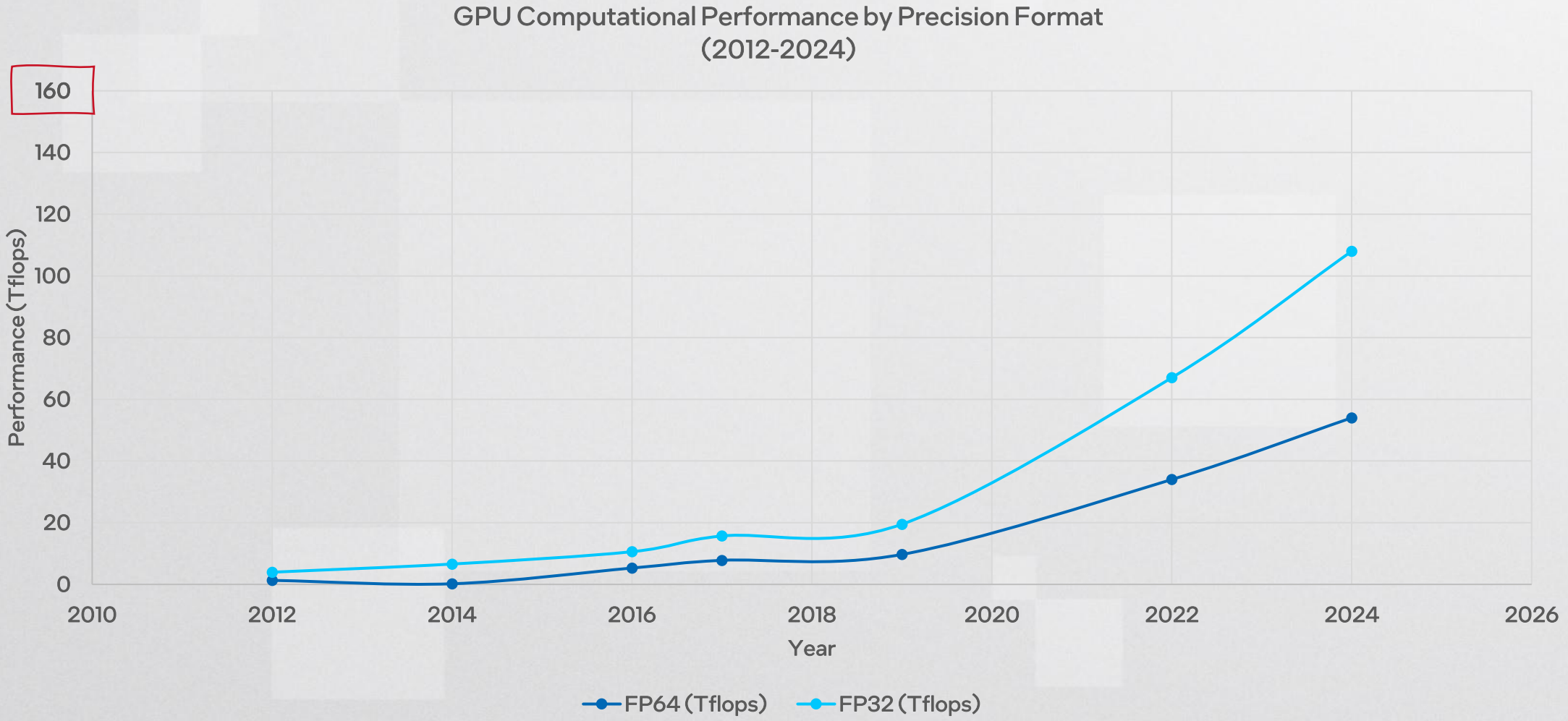


Intel® Core™ Ultra 200V series

> Utilizing oneAPI tools to optimize and run QMCPACK allows **the same code to run on both Aurora and my AI workstation** with Intel® Arc™ B580 GPU, the latter offering a cost-effective and ideal solution for my nightly validation and performance benchmark runs. As a developer, I appreciate the release of Intel® Arc™ B580 with oneAPI support as it **addresses an accessibility gap.**

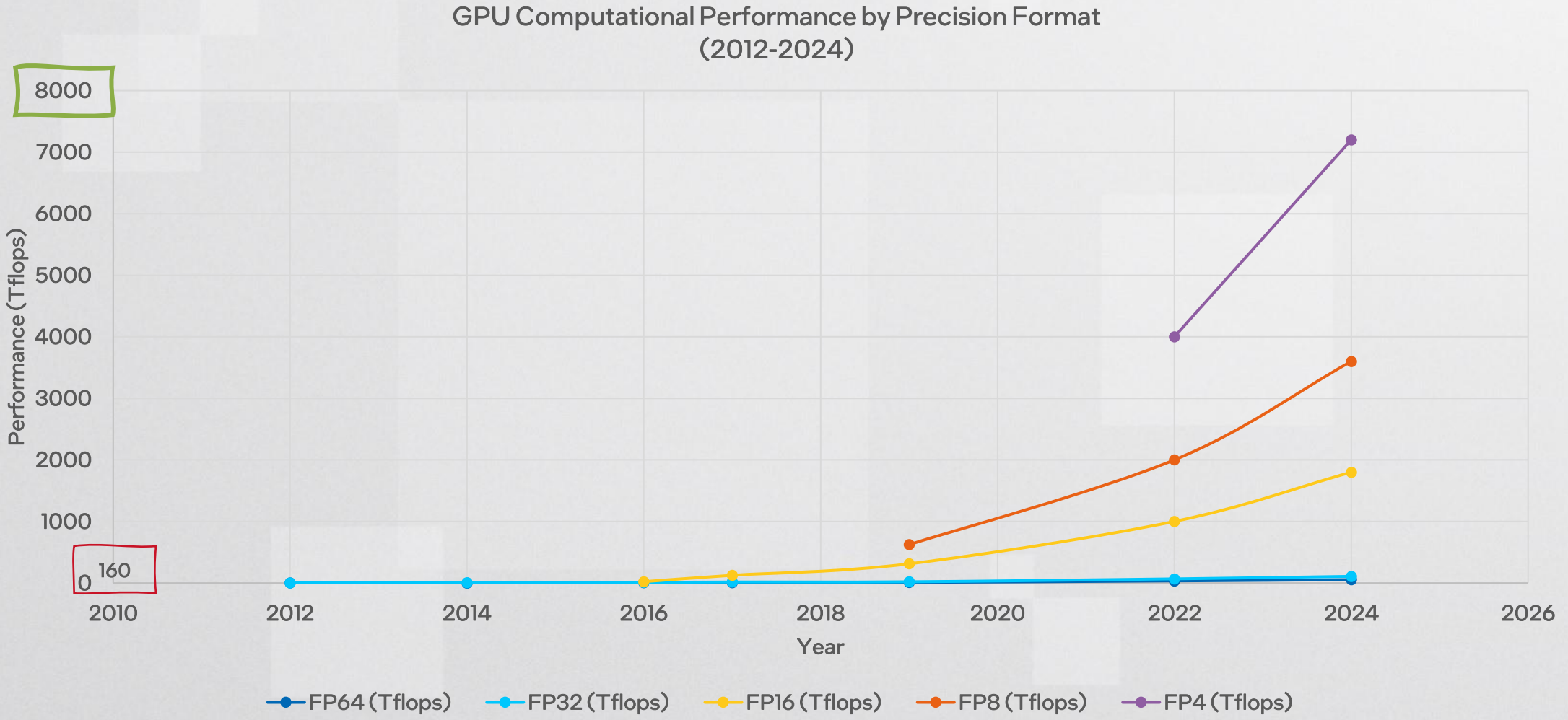
Ye Luo, Ph.D., Computational Science Division & Leadership Computing Facility, Argonne National Laboratory

What’s ahead?



Generated using Claude

Claude.ai (April 14, 2025). "show me an overtime plot peak FP64 flops in a high end GPU and compare that to fp32, fp16, fp8, fp4" [AI-generated table of values] Anthropic: https://claude.ai/chat/b4283459-bc58-40bf-8f84-fc5df62cc148



Generated using Claude

Claude.ai (April 14, 2025). "show me an overtime plot peak FP64 flops in a high end GPU and compare that to fp32, fp16, fp8, fp4" [AI-generated table of values] Anthropic: https://claude.ai/chat/b4283459-bc58-40bf-8f84-fc5df62cc148

Check out the *Using Low Precision* talk by Greg Henry @ 11:00am CDT

Low and Mixed precision support

oneMKL	Mixed precision via Alternate Compute Modes or Iterative Refinement Low precision support for GEMM, Vector Math, and other routines
oneDNN	Fast matrix multiply for many low precision data types
SYCL Joint Matrix	Language extension for matrix programming

SYCL Joint matrix

Check out the *Joint Matrix* talk by Dounia Khaldi @ 11:30am CDT

What is it?

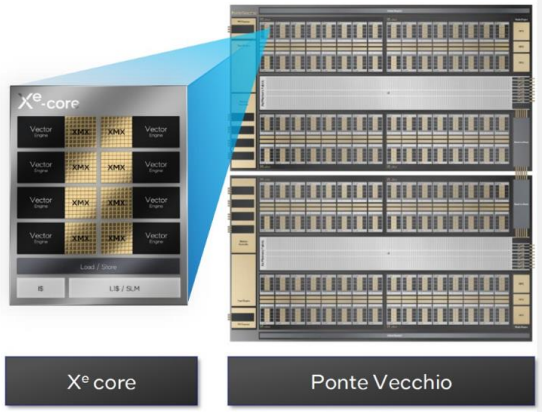
- A new SYCL extension for matrix programming
  - Part of Intel® oneAPI DPC++/C++ Compiler
  - Check out the specification on GitHub
- Direct, portable, and performant way to access matrix hardware

What is the benefit?

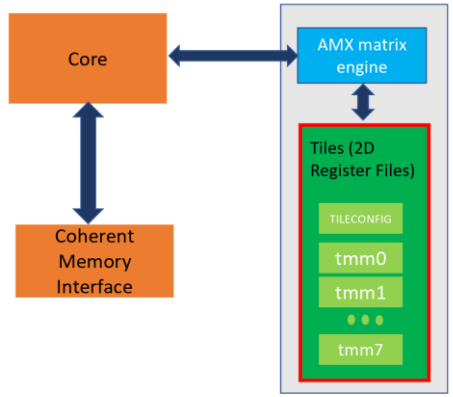
- Productivity: allows the developer to express their application for matrix hardware with minimal changes
- Performance: maps directly to low-level intrinsics
- Portability: write once, run everywhere

Where is it supported?

Intel® Xe Matrix Extensions for Intel® Arc™ Graphics



Intel® Advanced Matrix Extension in Intel® Xeon® 6



Nvidia* Tensor Cores

AMD* Matrix Cores

Join us!

CUTLASS

- Highly popular building blocks from NVidia for high-performance linear algebra
- C++ template library
- SYCL-enabled backend

6M downloads

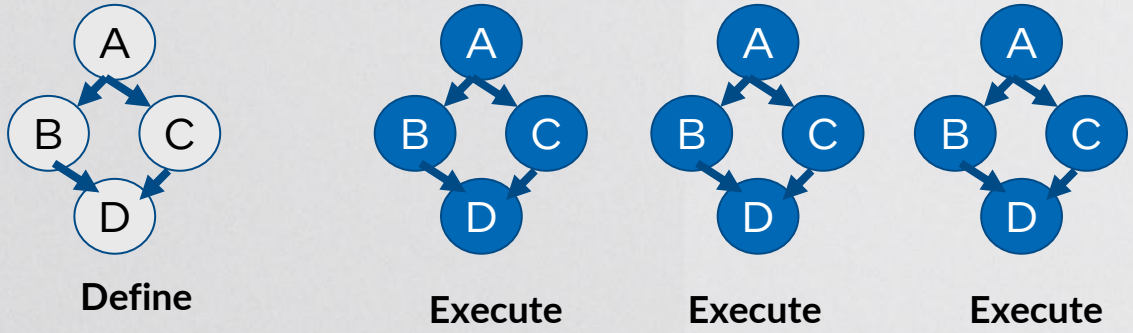
Used in 200+ projects

All Major DL Frameworks

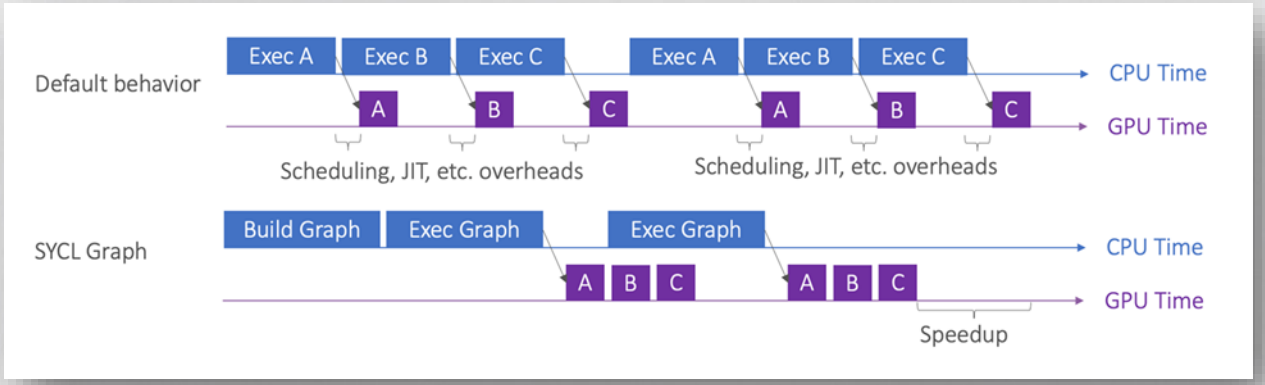
SYCL Graph

Check out the *SYCL Graph* talk by Pablo Reble @ 1:30pm CDT

- Define once, submit multiple times



- Reduce latency when submitting to device



- SYCL extension
  - Part of Intel® oneAPI DPC++/C++ Compiler
  - Available open-source on GitHub

- Supported in:

LLaMA C++

kokkos

FAST. FLEXIBLE. FREE. GROMACS

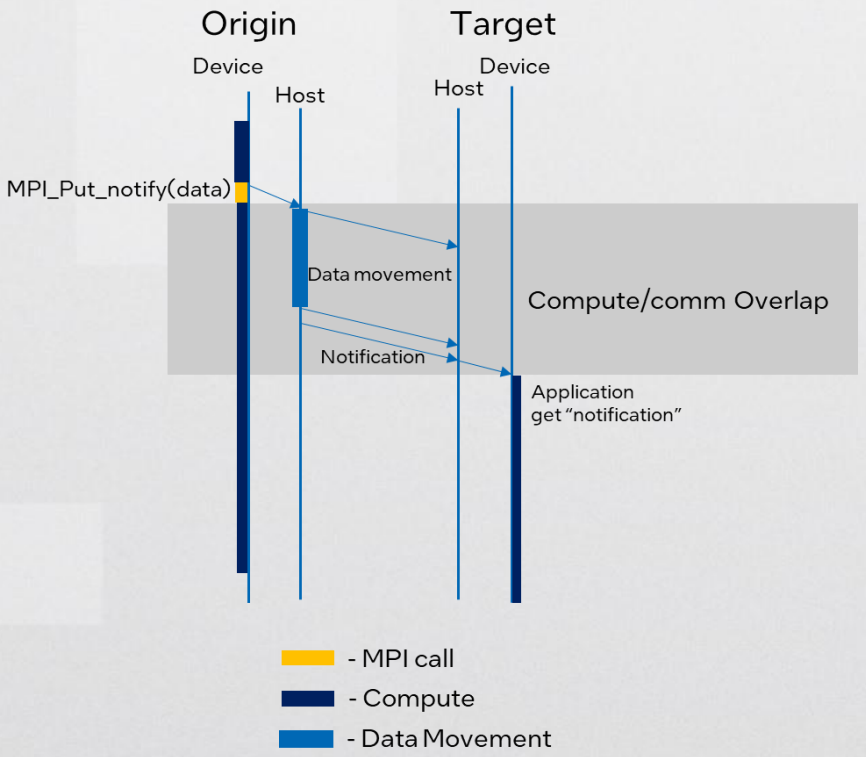
GPU-initiated communication with the Intel® MPI Library

Intel MPI Library

- Multi-fabric implementation of the Message Passing Interface specification
- Dynamic selection of optimal interconnect
- MPI 4.0 compliant

Device-initiated comms

- Triggered from GPU
- Execute on:
  - device (scale up)
  - host (scale out)
- Improve communication / computation overlap
- Supported on Intel and Nvidia GPUs



Intel Software Developer Tools …

… have a rich history of innovation

… are based on open standards

… give you choice

Join us in creating an open GPU programming ecosystem

Use the tools you know and love on today's integrated & discrete GPUs

From supercomputers to laptops!

Thank You

Disclaimers

Statements in this document that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.

All product plans and roadmaps are subject to change without notice.

Performance varies by use, configuration and other factors. Learn more on the Performance Index site. Intel technologies may require enabled hardware, software or service activation.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.


intel