Khronos Connects Software to Silicon

Open, royalty-free interoperability standards to harness the power of GPU, multiprocessor and XR hardware

3D graphics, augmented and virtual reality, parallel programming, inferencing and vision acceleration

Non-profit, member-driven standards organization, open to any company

Well-defined multi-company governance and IP Framework

Founded in 2000

>150 Members ~ 40% US, 30% Europe, 30% Asia
Khronos Active Initiatives

3D Graphics
- Desktop, Mobile and Web
  - Vulkan
  - ANARI
  - OpenGL ES
  - OpenGL
  - WebGL

3D Assets
- Authoring and Delivery
  - glTF
  - COLLADA
  - 3DCommerce

Portable XR
- Augmented and Virtual Reality
  - OpenXR

Parallel Computation
- Vision, Inferencing, Machine Learning
  - OpenCL
  - OpenVX
  - NNEF

Safety Critical APIs
- OpenGL sc
- Vulkan sc

This work is licensed under a Creative Commons Attribution 4.0 International License
Khronos Compute Acceleration Standards

Higher-level Languages and APIs
Streamlined development and performance portability

Single source C++ programming with compute acceleration

Graph-based vision and inferencing acceleration

Lower-level APIs
Direct Hardware Control

GPU rendering + compute acceleration

Intermediate Representation (IR) supporting parallel execution and graphics

Heterogeneous compute acceleration

Increasing industry interest in parallel compute acceleration to combat the ‘End of Moore’s Law’

SYCL and SPIR were originally OpenCL sub projects
SYCL 2020 Launch!
Open Standard for Single Source C++ Parallel Programming

SYCL 2020 is released after 3 years of intense work
Significant adoption in Embedded, Desktop and HPC markets
Improved programmability, smaller code size, faster performance
Based on C++17, backwards compatible with SYCL 1.2.1
Simplify porting of standard C++ applications to SYCL
Closer alignment and integration with ISO C++
Backend acceleration API independent

SYCL 2020 increases expressiveness and simplicity for modern C++ heterogeneous programming
SYCL 2020 Industry Momentum

SYCL (pronounced "sky-cl") is a royalty-free, cross-platform abstraction layer that builds and enhances OpenCL. It enables code written in a "single-source" style using a high level of abstraction, hiding platform dependencies and APIs.

NSITEXE, Kyoto Microcomputer, Codeplay Software Are Bringing Open Standards Programming to RISC-V Vector Processor for HPC and AI Systems

By Tara Hill

SYCL and DPC++ for Aurora

NERSC, ALCF, Codeplay Partner on SYCL for Next-Generation Supercomputers

Perlmutter

SYCL support growing from Embedded Systems through Desktops to Supercomputers
SYCL 2020 Major Features

- **Unified Shared Memory (USM)**
  - Code with pointers can work naturally without buffers or accessors

- **Parallel Reductions**
  - Added built-in reduction operation to avoid boilerplate code and achieve maximum performance on hardware with built-in reduction operation acceleration

- **Work group and subgroup algorithms**
  - Efficient parallel operations between work items

- **Class template argument deduction (CTAD) and template deduction guides**
  - Simplified class template instantiation

- **Simplified use of Accessors with a built-in reduction operation**
  - Reduces boilerplate code and streamlines the use of C++ software design patterns

- **Expanded interoperability**
  - Efficient acceleration by diverse backend acceleration APIs

- **SYCL atomic operations are now more closely aligned to standard C++ atomics**
  - Enhances parallel programming freedom
SYCL Single Source C++ Parallel Programming

- **C++ Libraries**
  - Standard C++ Application Code
  - ML Frameworks

- **SYCL Compiler**
  - SYCL Template Libraries
  - SYCL Compiler
  - CPU Compiler

- **OpenCL**
  - Accelerated code passed into device OpenCL compilers

**Complex ML frameworks can be directly compiled and accelerated**

**C++ templates and lambda functions separate host & accelerated device code**

**SYCL is ideal for accelerating larger C++-based engines and applications with performance portability**

- **One-MKL**
- **One-DNN**
- **OneDPC**
- **SYCL-BLAS**
- **SYCL-Eigen**
- **SYCL-DNN**
- **SYCL Parallel STL**

- **One-DNN**
- **OneDPC**
- **SYCL-BLAS**
- **SYCL-Eigen**
- **SYCL-DNN**
- **SYCL Parallel STL**

**C++ Kernel Fusion can give better performance on complex apps and libs than hand-coding**

**Accelerated code passed into device OpenCL compilers**
Parallel Industry Initiatives

C++11 2011
OpenCL 1.2
OpenCL C Kernel Language

C++14
SYCL 1.2
C++11 Single source programming

C++17 2015
OpenCL
OpenCL 2.1
SPIR-V in Core

SYCL 1.2.1
C++11 Single source programming

C++20 2017
OpenCL
OpenCL 2.2

SYCL 2020
C++17 Single source programming
Many backend options

C++23 2020
OpenCL 3.0

SYCL 202X
C++20 Single source programming
Many backend options

C++23
OpenCL

C++17
SPIR

C++14
SYCL

C++11
Parallel Industry Initiatives

SYCL Implementations in Development

- SYCL, OpenCL and SPIR-V, as open industry standards, enable flexible integration and deployment of multiple acceleration technologies.
- SYCL enables Khronos to influence ISO C++ to (eventually) support heterogeneous compute.

Multiple Backends in Development
SYCL beginning to be supported on multiple low-level APIs in addition to OpenCL e.g., ROCm and CUDA
For more information: http://sycl.tech
SYCL in Embedded Systems

- Networks trained on high-end desktop and cloud systems
- Applications link to compiled inferencing code or call vision/inferencing API
- Diverse Embedded Hardware: Multi-core CPUs, GPUs, DSPs, FPGAs, Tensor Cores
  * Vulkan only runs on GPUs

Open industry standards, enable flexible integration and deployment of multiple acceleration technologies

This work is licensed under a Creative Commons Attribution 4.0 International License
SYCL in HPC/Supercomputers

Three Pillars of Science Problem

Simulation
HPC Languages
Solver Libraries, Parallel RT

Data
Productivity Languages
Big Data Stack, Stats Lib, Databases

Learning
Productivity Languages
Deep Learning, Linear Alg, ML

Need Languages that allow control of these Data Issues
Set Data affinity, Data Layout, Data movement, Data Locality, highly Parameterized Code and dynamically compose the algorithms (C++ templates, parallel STL, inlining and fusion, abstractions)

Libraries augment compiler optimizations for Performance Portable programs

Use open standards to run Performance Portable code on new generation, or different vendor’s, hardware with compiler optimization, explicit parametrization and dynamically composed algorithm

OpenMP for C and Fortran
CUDA/pthreads/OpenACC/OpenCL
C++ Application uses SYCL, Kokkos, Raja

Math, ML, Data Libraries; C++ Std, C, Python Libraries

Today’s Supercomputing Development Workflow needs knowledge of system architecture and tools that control data

Choose Algorithm for target
Implement and Test Algorithm
Optimize Algorithm


This work is licensed under a Creative Commons Attribution 4.0 International License
Enabling Industry Engagement

- SYCL working group values industry feedback
  - https://community.khronos.org/c/sycl
  - https://sycl.tech
- What features would you like in future SYCL versions?

Public contributions to Specification, Conformance Tests and software
https://github.com/KhronosGroup/SYCL-CTS
https://github.com/KhronosGroup/SYCL-Doct
https://github.com/KhronosGroup/SYCL-Shared
https://github.com/KhronosGroup/SYCL-Registry
https://github.com/KhronosGroup/SyctParallelSTL

Invited Experts
https://www.khronos.org/advisors/

Khrnos SYCL Forums, Slack Channels, Stackoverflow, reddit, and SYCL.tech
Open to all!
https://community.khronos.org/www.khr.io/slack
https://app.slack.com/client/TDMDFS87M/CE9UX4CHG
https://community.khronos.org/c/sycl/
https://stackoverflow.com/questions/tagged/sycl
https://www.reddit.com/r/sycl
https://github.com/codeplaysoftware/syclacademy
https://sycl.tech/

Khrnos GitHub
Contribute to SYCL open source specs, CTS, tools and ecosystem

SYCL Advisory Panels

SYCL Working Group

Khrnos members
https://www.khronos.org/members/
https://www.khronos.org/registry/SYCL/
Thank You! Questions?

- **Khronos SYCL** is creating a cutting-edge royalty-free open standard
  - For C++ Heterogeneous compute, vision, inferencing acceleration
  - [https://www.khronos.org/sycl](https://www.khronos.org/sycl)

- Any entity/individual is welcome to join Khronos to influence SYCL’s evolution
  - [https://www.khronos.org/members](https://www.khronos.org/members)

- Join **IWOCL/SYCLcon** Conference
  - Wednesday Apr 27-29, 2021

- **Michael Wong**
  - michael@codeplay.com | wongmichael.com/about

---

**Benefits of Khronos Membership**

- Gather industry requirements for future open standards
- Draft Specifications Confidential to Khronos members
- Publicly Release Specifications and Conformance Tests
- Gain early insights into industry trends and directions
- Influence the design and direction of key open standards that will drive your business
- Accelerate your time-to-market with early access to specification drafts
- Network with domain experts from diverse companies in your industry
- State-of-the-art IP Framework protects your Intellectual Property
- Enhance your company reputation as an industry leader through Khronos participation