SYCL State of the Union Keynote SYCLCon 2021

Specification Release

Michael Wong
SYCL WG Chair
Codeplay Distinguished Engineer
ISOCPP Director & VP
ISO C++ Directions Group
michael@codeplay.com | wongmichael.com/about

SYCL 2020 is here!

Open Standard for Single Source C++ Parallel Heterogeneous 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

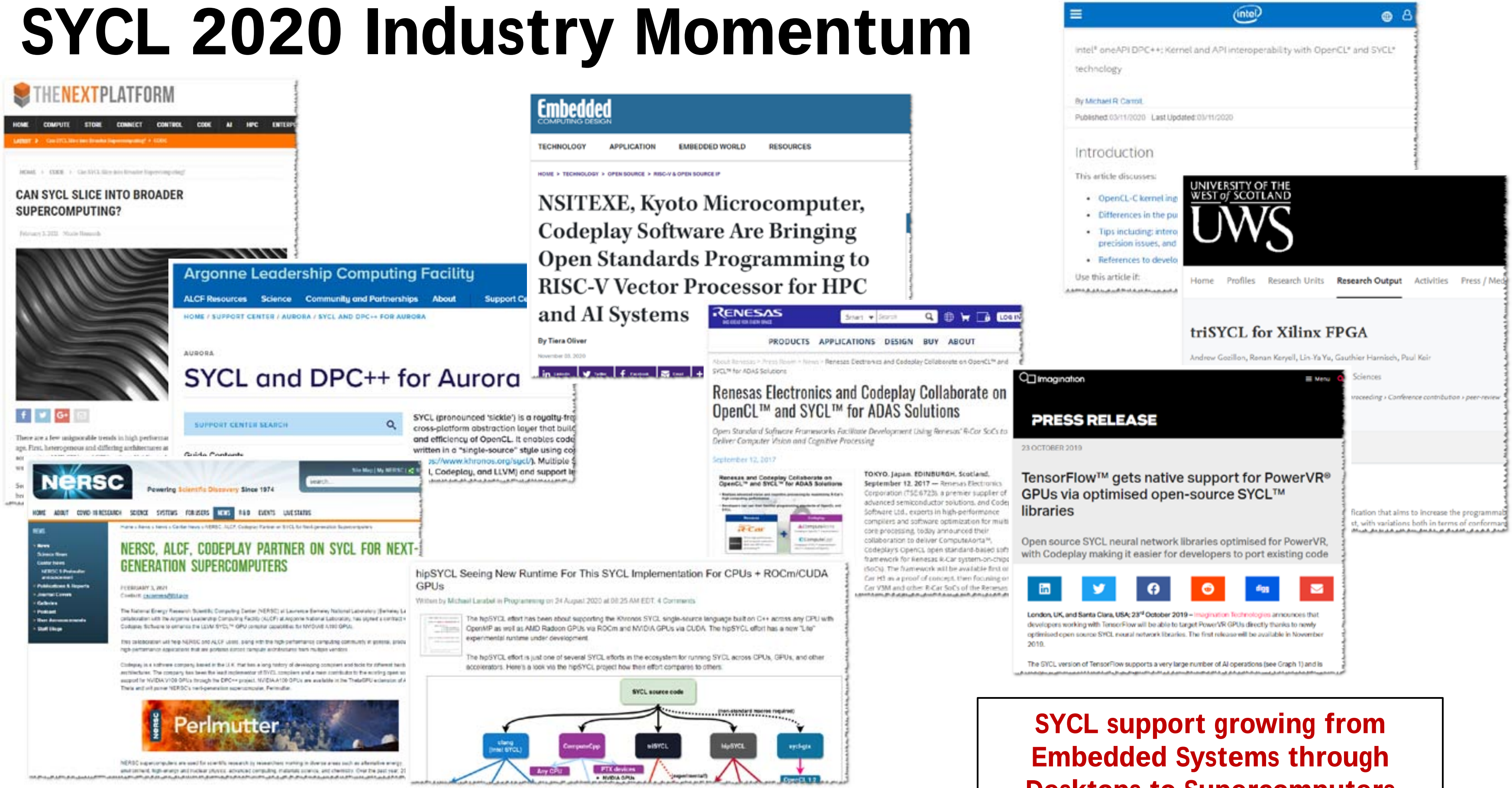
Simplify porting of standard C++ applications to SYCL

Closer alignment and integration with ISO C++

Multiple Backend acceleration and API independent

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

SYCL 2020 Industry Momentum



THENEXTPLATFORM

HOME COMPUTE STORE CONNECT CONTROL CODE AI HPC ENTERPR

CAN SYCL SLICE INTO BROADER SUPERCOMPUTING?

Argonne Leadership Computing Facility

ALCF Resources Science Community and Partnerships About Support Ce

HOME / SUPPORT CENTER / AURORA / SYCL AND DPC++ FOR AURORA

AURORA

SYCL and DPC++ for Aurora

SUPPORT CENTER SEARCH

SYCL (pronounced 'sickle') is a royalty-fre cross-platform abstraction layer that build and efficiency of OpenCL. It enables code written in a "single-source" style using co ps://www.khronos.org/sycl/). Multiple l, Codeplay, and LLVM) and support fo

Embedded COMPUTING DESIGN

TECHNOLOGY APPLICATION EMBEDDED WORLD RESOURCES

HOME > TECHNOLOGY > OPEN SOURCE > RISC-V & OPEN SOURCE IP

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

By Tiera Oliver

intel

Intel® oneAPI DPC++: Kernel and API interoperability with OpenCL* and SYCL* technology

By Michael R Carroll

Published: 03/11/2020 Last Updated: 03/11/2020

Introduction

This article discusses:

- OpenCL-C kernel ing
- Differences in the pu
- Tips including: intero precision issues, and
- References to develo

Use this article if:

UNIVERSITY OF THE WEST of SCOTLAND UWS

Home Profiles Research Units Research Output Activities Press / Med

triSYCL for Xilinx FPGA

Andrew Gozillon, Ronan Keryell, Lin-Ya Yu, Gauthier Harnisch, Paul Keir

RENESAS

PRODUCTS APPLICATIONS DESIGN BUY ABOUT

Renesas Electronics and Codeplay Collaborate on OpenCL™ and SYCL™ for ADAS Solutions

Open Standard Software Frameworks Facilitate Development Using Renesas' R-Car SoCs to Deliver Computer Vision and Cognitive Processing

September 12, 2017

NERSC Powering Scientific Discovery Since 1974

NERSC, ALCF, CODEPLAY PARTNER ON SYCL FOR NEXT-GENERATION SUPERCOMPUTERS

FEBRUARY 3, 2021

Perlmutter

hipSYCL Seeing New Runtime For This SYCL Implementation For CPUs + ROCm/CUDA GPUs

Written by Michael Larabel in Programming on 24 August 2020 at 08:25 AM EDT. 4 Comments

The hipSYCL effort has been about supporting the Khronos SYCL single-source language built on C++ across any CPU with OpenMP as well as AMD Radeon GPUs via ROCm and NVIDIA GPUs via CUDA. The hipSYCL effort has a new "Lite" experimental runtime under development.

The hipSYCL effort is just one of several SYCL efforts in the ecosystem for running SYCL across CPUs, GPUs, and other accelerators. Here's a look via the hipSYCL project how their effort compares to others:

Imagination

PRESS RELEASE

23 OCTOBER 2019

TensorFlow™ gets native support for PowerVR® GPUs via optimised open-source SYCL™ libraries

Open source SYCL neural network libraries optimised for PowerVR, with Codeplay making it easier for developers to port existing code

London, UK, and Santa Clara, USA; 23rd October 2019 – Imagination Technologies announces that developers working with TensorFlow will be able to target PowerVR GPUs directly thanks to newly optimised open source SYCL neural network libraries. The first release will be available in November 2019.

The SYCL version of TensorFlow supports a very large number of AI operations (see Graph 1) and is

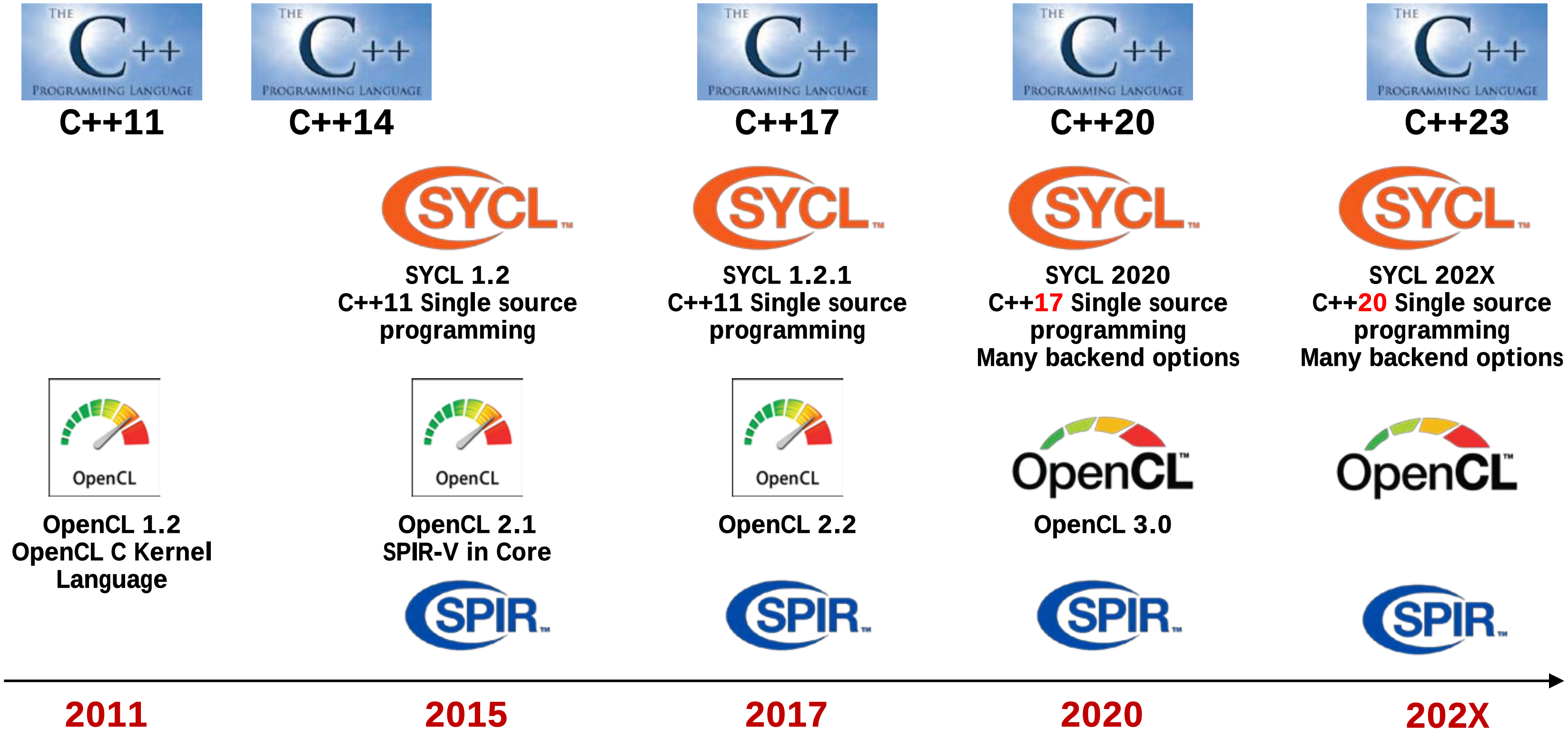
https://www.alcf.anl.gov/support-center/aurora/sycl-and-dpc-aurora
https://www.embeddedcomputing.com/technology/open-source/risc-v-open-source-ip/nsitexe-kyoto-microcomputer-and-codeplay-software-are-bringing-open-standards-programming-to-risc-v-vector-processor-for-hpc-and-ai-systems
https://www.nextplatform.com/2021/02/03/can-sycl-slice-into-broader-supercomputing/
https://www.phoronix.com/scan.php?page=news_item&px=hipSYCL-New-Lite-Runtime
https://software.intel.com/content/www/us/en/develop/articles/interoperability-dpcpp-sycl-opencl.html
https://www.renesas.com/br/en/about/press-room/renesas-electronics-and-codeplay-collaborate-opencl-and-sycl-adas-solutions
https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/2021/nersc-alcf-codeplay-partner-on-sycl-for-next-generation-supercomputers/
https://research-portal.uws.ac.uk/en/publications/trisycl-for-xilinx-fpga
https://www.imaginationtech.com/news/press-release/tensorflow-gets-native-support-for-powervr-gpus-via-optimised-open-source-sycl-libraries/

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
  - Simplifies porting from most code (e.g. CUDA, C++)
- **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

Parallel Industry Initiatives



C++11

C++14

SYCL 1.2
C++11 Single source programming

C++17

SYCL 1.2.1
C++11 Single source programming

C++20

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

C++23

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

OpenCL 1.2
OpenCL C Kernel Language

OpenCL 2.1
SPIR-V in Core

OpenCL 2.2

OpenCL 3.0

2011

2015

2017

2020

202X

SYCL Evolution

SYCL 2020 compared with SYCL 1.2.1

Easier to integrate with C++17 (CTAD, Deduction Guides...)
Less verbose, smaller code size, simplify patterns
Backend independent
Multiple object archives aka modules simplify interoperability
Ease porting C++ applications to SYCL
Enable capabilities to improve programmability
Backwards compatible but minor API break based on user feedback

SYCLCon 2020 Talks and Events

- SYCL, DPC++, SPUs, oneAPI - a View from Intel by James Reinders
- oneAPI Developer Summit Monday Apr 26, Biagio Cosenza, Peter Zuzek, Steffen Larsen
- Hands on SYCL Tutorial Tuesday Apr 27 by Rod Burns and SYCL team
- Sylkan: Towards a Vulkan Compute Target Platform for SYCL by Peter Thorman
- Performance-Portable Distributed K-Nearest neighbours using Locality-Sensitive Hashing and SYCL by Marcel Breyer
- Toward Performance Portability of Highly Parametrizable TRSM Algorithm Using SYCL by Thales Sabino
- On Measuring the Maturity of SYCL implementations by Tracking Historical Performance improvements by Wei-Chen Lin
- Experiences Supporting DPC++ in AMRex by Sravani Konda
- Developing Medical Imaging Applications Across GPU, FPGA, and CPU using oneAPi
- hipSYCL in 2021: Peculiarities, Unique Features and SYCL 2020 by Aksel Alpay
- Experiences with Adding SYCL Support to GROMACS by Andrewy Alekseenko
- Extending DPC++ with SUpport for Huawei Ascend AI Chipset
- Toward a Better SYCL Memory Consistency Model by Ben Ashb
- Bringing SYCL to A100 Ampere Architecture on Perlmutter Steffen Larsen and LBNL
- SYCL and OpenCL Meet Challenges of Functional Safety by illya Rudkin
- Enabling OpenCL and SYCL for RISC-V processors by Colin Davidson, Aidan Dodds
- SYCL Panel Thursday Apr 29

SYCL Future Roadmap (MAY CHANGE)

SYCL 2020

Over 40 Selected Features for SYCL 2020

Unified Shared Memory)
Parallel Reductions adds a built in reduction operation
Work-group and sub-group algorithms
Improvements to atomic operations
Class template argument deduction (CTAD) and deduction guides
Simplification of accessors
Expanded interoperability with different backends
Extension mechanism
Address spaces
Vector rework
Specialization Constants
...

Improving Software Ecosystem

Books, Tutorials, Tool, libraries, GitHub

Expanding Implementation

DPC++
ComputeCpp
triSYCL
hipSYCL
neoSYCL

Regular Maintenance Updates

Spec clarifications, formatting and bug fixes
https://www.khronos.org/registry/SYCL/

Repeat The Cycle every 1.5-3 years

NEXT

Integration of successful Extensions plus new Core functionality

Conformance Tests

Working on Implementations

Future SYCL NEXT Proposals

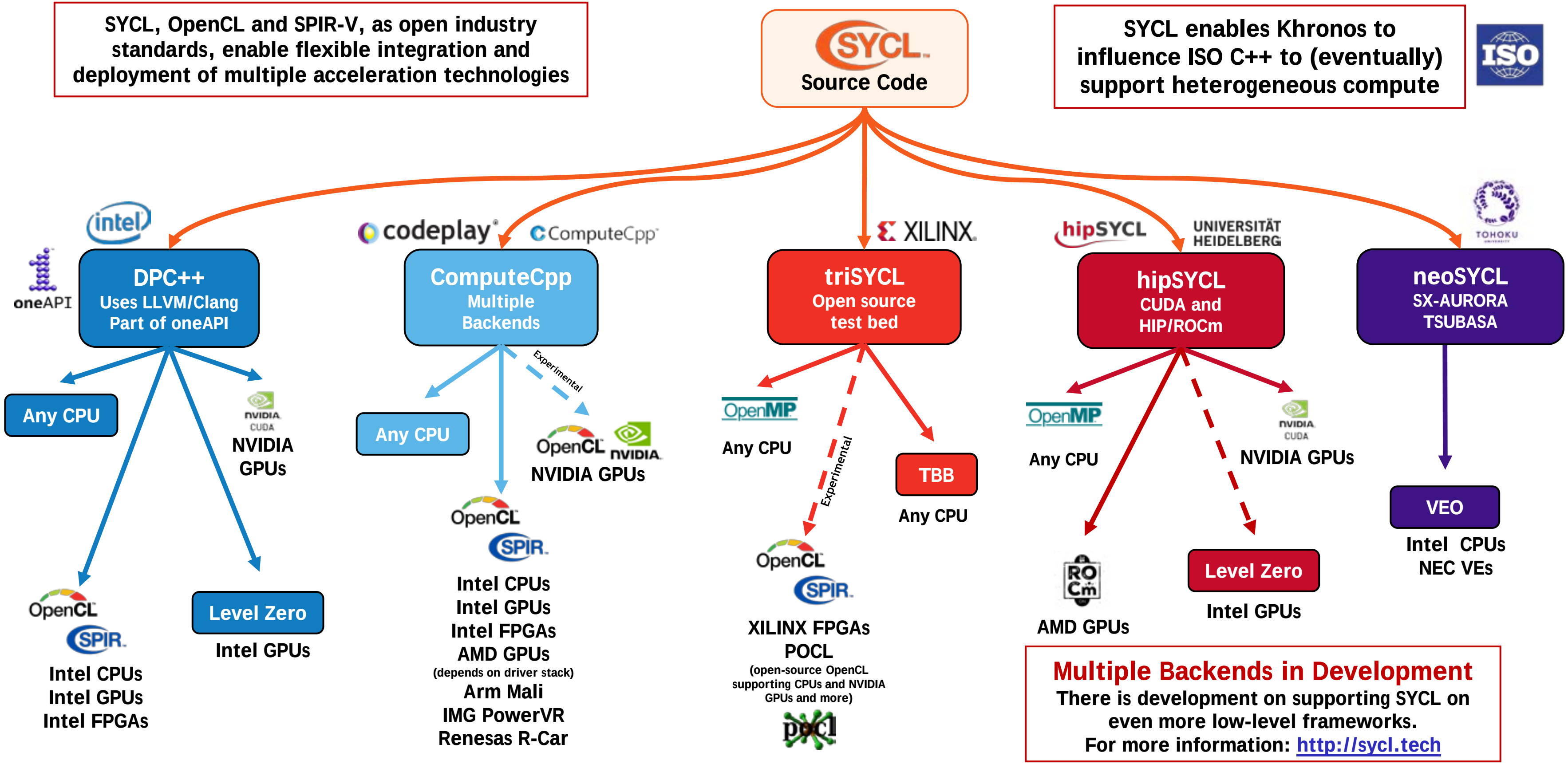
...

Converge SYCL with ISO C++ and continue to support OpenCL to deploy on more devices
CPU
GPU
FPGA
AI processors
Custom Processors

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



SYCL Source Code

DPC++ Uses LLVM/Clang Part of oneAPI (intel, oneAPI)
- Any CPU
- NVIDIA CUDA: NVIDIA GPUs
- OpenCL SPIR: Intel CPUs, Intel GPUs, Intel FPGAs
- Level Zero: Intel GPUs

ComputeCpp Multiple Backends (codeplay, ComputeCpp)
- Any CPU
- Experimental: OpenCL NVIDIA: NVIDIA GPUs
- OpenCL SPIR: Intel CPUs, Intel GPUs, Intel FPGAs, AMD GPUs (depends on driver stack), Arm Mali, IMG PowerVR, Renesas R-Car

triSYCL Open source test bed (XILINX)
- OpenMP: Any CPU
- TBB: Any CPU
- Experimental: OpenCL SPIR: XILINX FPGAs, POCL (open-source OpenCL supporting CPUs and NVIDIA GPUs and more) pocl

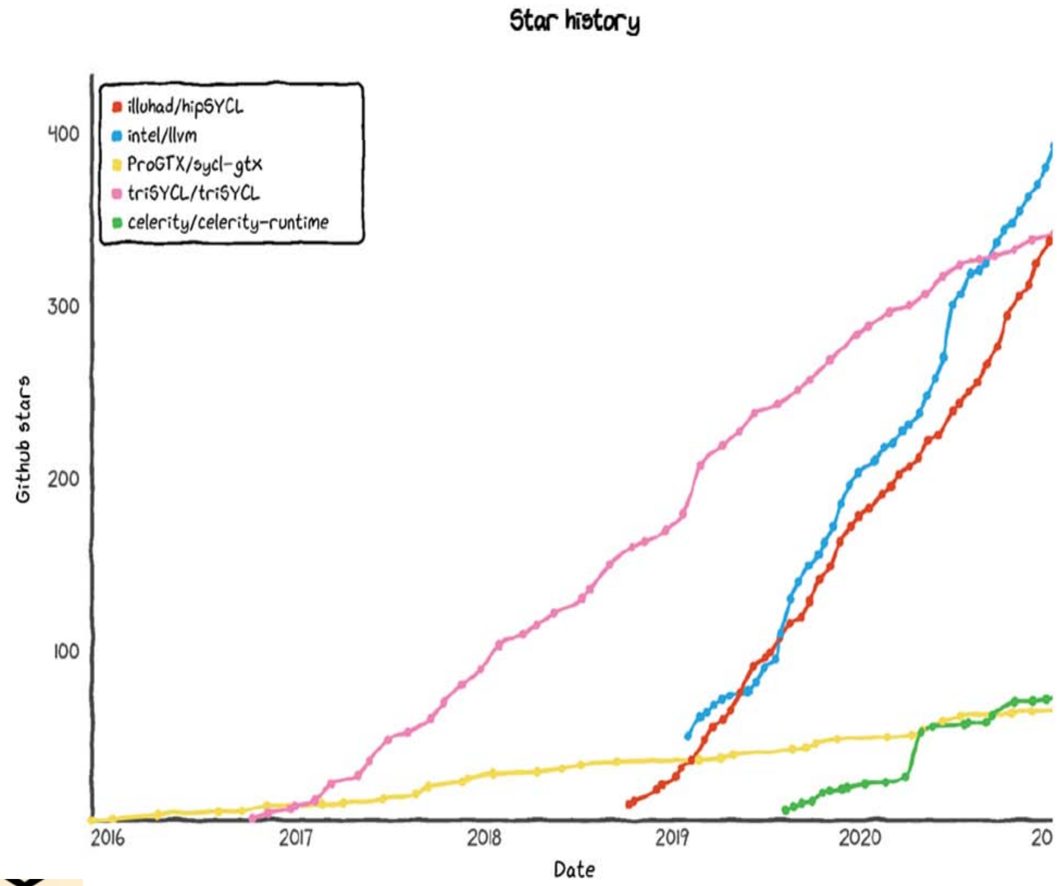
hipSYCL CUDA and HIP/ROCm (hipSYCL, UNIVERSITÄT HEIDELBERG)
- OpenMP: Any CPU
- ROCm: AMD GPUs
- NVIDIA CUDA: NVIDIA GPUs
- Level Zero: Intel GPUs

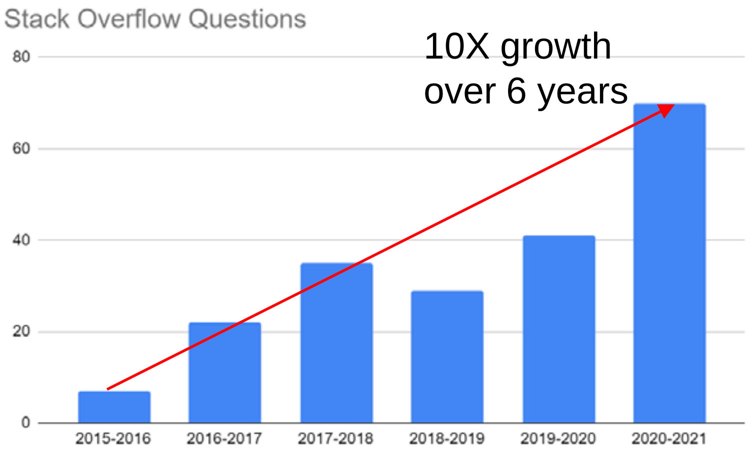
neoSYCL SX-AURORA TSUBASA (TOHOKU)
- VEO: Intel CPUs, NEC VEs

Multiple Backends in Development

There is development on supporting SYCL on even more low-level frameworks.
For more information: http://sycl.tech

SYCL user and developer Growth





SYCL Ecosystem, Research and Benchmarks

Implementations

neoSYCL
SX-AURORA TSUBASA

oneAPI

oneAPI
DATA PARALLEL C++
Performant, Productive Programming for Heterogeneous Architecture

ComputeCpp™

hipSYCL

Research

Celerity
High-level C++ for Accelerator Clusters

ECP
PROJECT

kokkos

Taskflow: A General-purpose Parallel and Heterogeneous Task Programming System using Modern C++

Dr. Tsung-Wei (TW) Huang
Department of Electrical and Computer Engineering
University of Utah, Salt Lake City, UT
https://taskflow.github.io/

RAJA

2 sessions

Grid: A High-Performance and Portable Code for Quantum Chromodynamics. [S32914]
Wednesday, Apr 14 | 9:00 AM - 9:40 AM EST
Peter Boyle, Professor, University of Edinburgh and Brookhaven National Laboratory

SYCL 2020 Programming Environment for A100 [S32766]
Brandon Cook, Application Performance Specialist, Lawrence Berkeley National Laboratory

ATLAS EXPERIMENT

alpaka

Benchmarks/Books

Direct Programming Benchmark

Data Parallel C++

Mastering DPC++ for Programming of Heterogeneous Systems using C++ and SYCL

James Reinders
Ben Ashbaugh
James Brodman
Michael Kinsner
John Pennycook
Xinmin Tian

Apress open

FREE eBook

Linear Algebra Libraries

Machine Learning Libraries and Parallel Acceleration Frameworks

BLAS	FFT	Math	RAND
SYCLBLAS oneMKL	oneMKL	oneMKL	oneMKL
SOLVER	**SPARSE**	**TENSOR**	**STL**
oneMKL	oneMKL	SYCL-DNN Eigen oneDNN TensorFlow	SYCL Parallel STL oneDPL

SYCL-Bench

STREAM HPC

Argonne NATIONAL LABORATORY

QUALCOMM

arm

AMD

SYCL™

universität innsbruck

intel

XILINX

codeplay

TOHOKU UNIVERSITY

University of BRISTOL

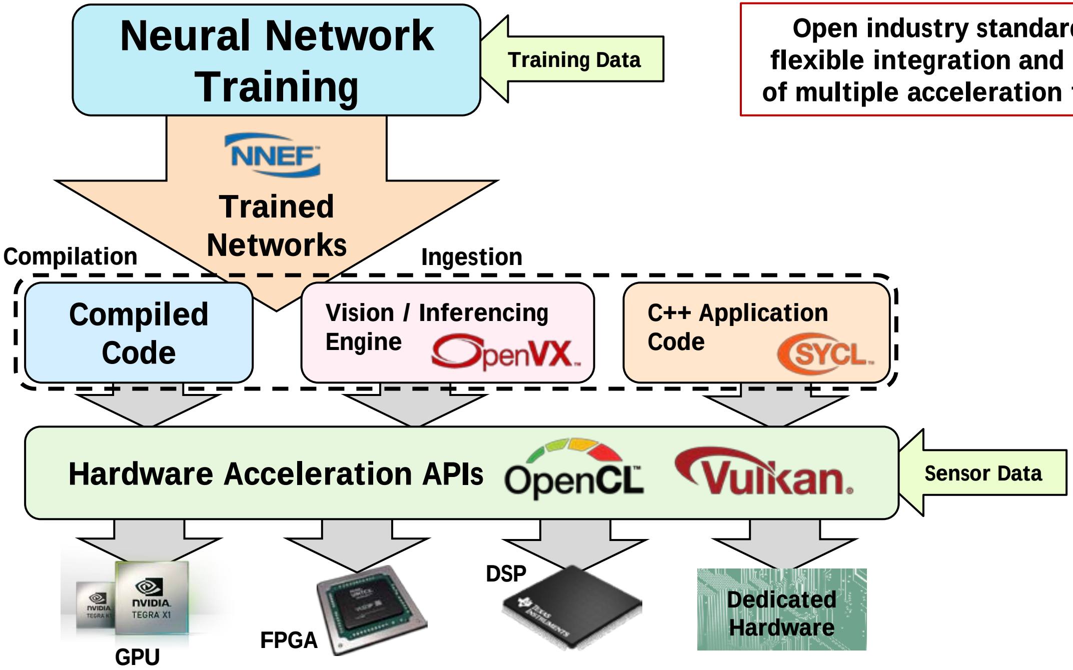
Working Group Members

SYCL in Embedded Systems, Automotive, and AI

Networks trained on high-end desktop and cloud systems

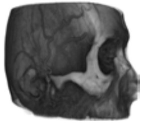
Applications link to compiled inferencing code or call vision/inferencing API

Diverse Embedded Hardware
Multi-core CPUs, GPUs
DSPs, FPGAs, Tensor Cores
* Vulkan only runs on GPUs

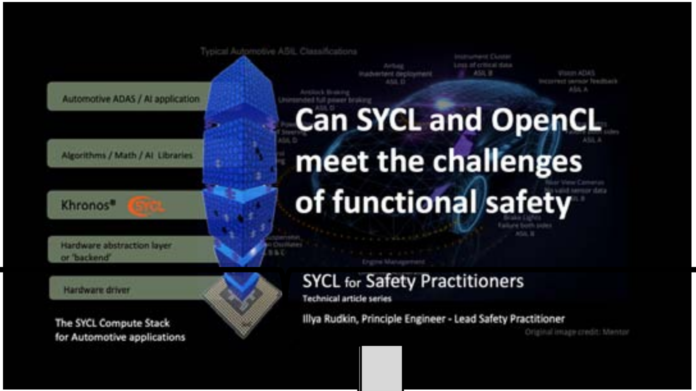


Open industry standards, enable flexible integration and deployment of multiple acceleration technologies

Safety Critical API Evolution



New Generation Safety Critical APIs for Graphics, Compute and Display



Typical Automotive ASIL Classifications
Automotive ADAS / AI application
Algorithms / Math / AI Libraries
Khronos® SYCL
Hardware abstraction layer or 'backend'
Hardware driver

Can SYCL and OpenCL meet the challenges of functional safety

The SYCL Compute Stack for Automotive applications

SYCL for Safety Practitioners

Technical article series

Illya Rudkin, Principle Engineer - Lead Safety Practitioner

Original image credit: Mentor

OpenCL

OpenCL and SYCL SC work will minimize API surface area，reduce ambiguity, UB, increase determinism

SYCL™

Rendering Compute Display

Industry Need
for GPU Acceleration APIs designed to ease system safety certification is increasing
ISO 26262 / ASIL-D

RISC-V

SCSC
FOR EVERYONE WORKING IN SYSTEM SAFETY
WWW.SCSC.UK

ISO International Organization for Standardization

UNECE WP.29

ISO 26262

ISO/PAS 21448

UL 4600

MISRA

SAE INTERNATIONAL.

ISO/IEC JTC 1/SC 42

Artificial intelligence

IBM

- ISO/IEC AWI TR 20547-1 [illegible]
  Information technology — Big data reference architecture — Part 1: Framework and application process
- ISO/IEC TR 20547-2:2018
  Information technology — Big data reference architecture — Part 2: Use cases and derived requirements
- ISO/IEC DIS 20547-3 [illegible]
  Information technology — Big data reference architecture — Part 3: Reference architecture
- ISO/IEC TR 20547-5:2018
  Information technology — Big data reference architecture — Part 5: Standards roadmap
- ISO/IEC AWI 22989 [illegible]
  Artificial Intelligence Concepts and Terminology
- ISO/IEC AWI 23053 [illegible]
  Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

Source: [illegible]

Embedded/Automotive/AI/Safety

"For Renesas, SYCL is a key enabler for automotive ADAS/AD software developers ….," **said Cyril Cordoba, Director of ADAS Segment Marketing Department, Renesas**.

"Xilinx is excited about the progress achieved with SYCL 2020," **said Ralph Wittig, fellow, Xilinx.**

"NSITEXE supports the SYCL 2020 technology, which is gaining attention in embedded applications," **said Hideki Sugimoto, CTO, NSITEXE, Inc.** "

"Imagination recognises the benefit of SYCL across multiple markets. Our software stacks have been designed to improve SYCL performance, enabling a straightforward path to exploit the teraflops of compute performance in our latest IP," **said Mark Butler, Vice President of Software Engineering**, **Imagination Technologies.**

SYCL support from embedded systems, through desktops to supercomputers

SYCL in HPC/Supercomputers

Simulation
HPC Languages
Solver Libraries, Parallel RT

Data
Productivity Languages
Big Data Stack, Stats Lib, Databases

Learning
Productivity Languages
Deep Learning, Linear Alg, ML

Three Pillars of Science Problem

OpenMP for C and Fortran

CUDA/pthreads/ OpenACC/OpenCL

C++ Application uses SYCL, Kokkos, Raja

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

2021

SYCL

2020

SYCL

2021

2021

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

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

Based on IWOCL/SYCLCon 2020 keynote Hal FInkel: https://www.iwocl.org/wp-content/uploads/iwocl-syclcon-2020-finkel-keynote-slides.pdf

Exascale computing

Argonne NATIONAL LABORATORY

“Our users will benefit from features in the SYCL 2020 specification. New features, such as support for unified memory (USM) and reductions, are important capabilities for programming high-performance-computing hardware. ...” **said Nevin Liber, computer scientist, Argonne National Laboratory’s Leadership Computing Facility**

CINECA

“At Cineca, based on our experience, we confirm the value that SYCL is bringing to the development of high-performance computing in a hybrid environment. ...” **said Sanzio Bassini, director of supercomputing, Application Innovation Dept, Cineca.**

SYCL support from embedded systems, through desktops to supercomputers

HPC Computing

universität innsbruck

UNIVERSITÀ DEGLI STUDI DI SALERNO

. "...As co-developers of the Celerity project, together with the University of Salerno, we are welcoming these changes and look forward to applying them within distributed-memory research and industry applications, for example as part of the recently launched EuroHPC LIGATE project." **said Thomas Fahringer, head of the Distributed and Parallel Systems Group at the University of Innsbruck**

NERSC

"... we see modern C++ language-based approaches to accelerator programming, such as SYCL, as an important component of our programming environment offering for users of Perlmutter," **said Brandon Cook, application performance specialist at NERSC.**

SYCL support from embedded systems, through desktops to supercomputers

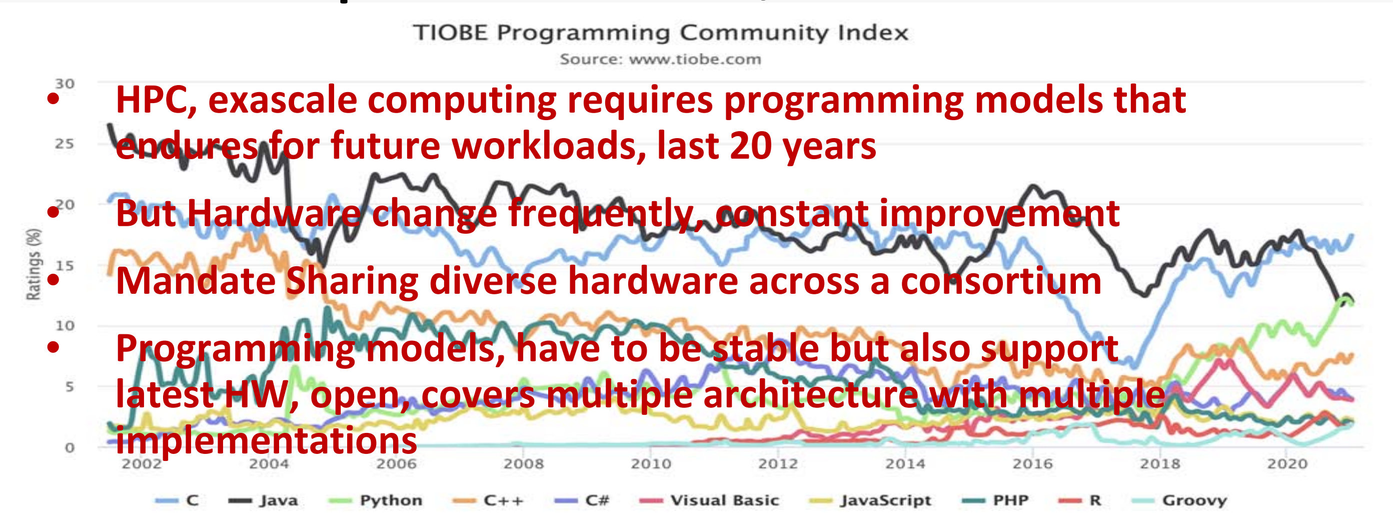
intel

"The SYCL 2020 final specification brings significant features to the industry that enable C++ developers to more productively build high-performance heterogeneous applications with unified programming across XPU architectures," **said Jeff McVeigh, Intel vice president, Datacenter XPU Products and Solutions.**

What now?

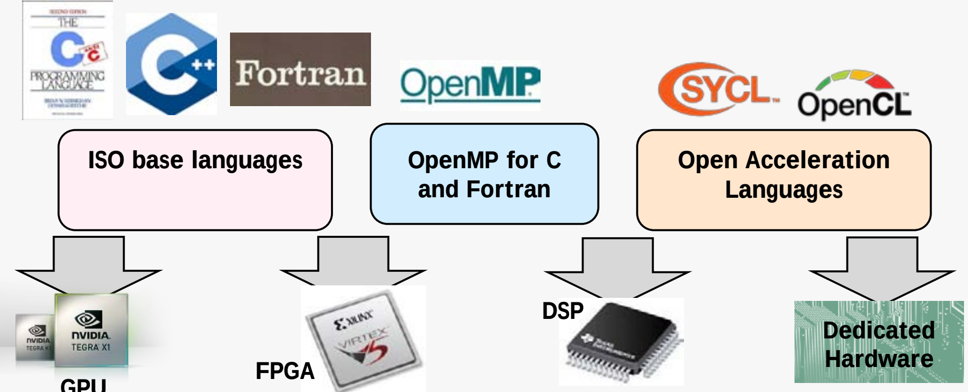
Deep Dive into HPC future

When I was OpenMP CEO, I learned



- HPC, exascale computing requires programming models that endures for future workloads, last 20 years
- But Hardware change frequently, constant improvement
- Mandate Sharing diverse hardware across a consortium
- Programming models, have to be stable but also support latest HW, open, covers multiple architecture with multiple implementations

OpenMP is great for C and Fortran



ISO base languages

OpenMP for C and Fortran

Open Acceleration Languages

GPU

FPGA

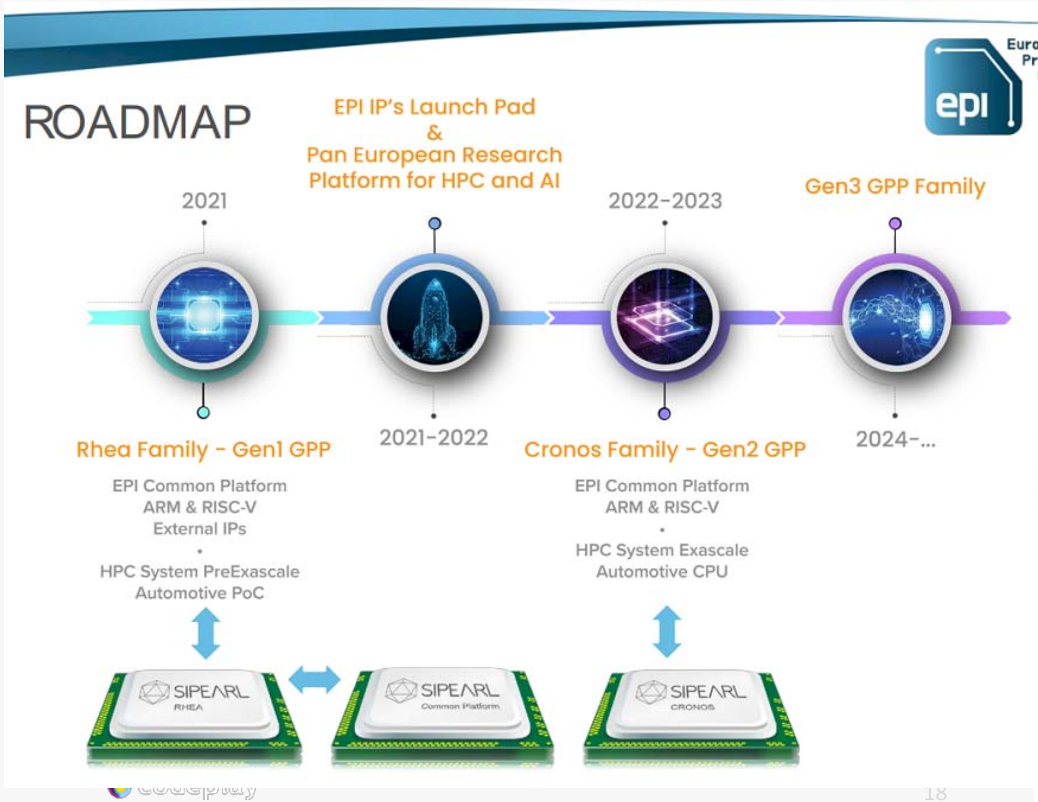
DSP

Dedicated Hardware

SYCL is great for modern C++, AI, Automotive

Here are some opportunities for HPC growth across Europe, Asia

What about Europe? EPI, ARM and RISC-V RVV



European Processor Initiative

epi

ROADMAP

2021

Rhea Family – Gen1 GPP

EPI Common Platform
ARM & RISC-V
External IPs
·
HPC System PreExascale
Automotive PoC

EPI IP's Launch Pad
&
Pan European Research
Platform for HPC and AI

2021-2022

2022-2023

Cronos Family – Gen2 GPP

EPI Common Platform
ARM & RISC-V
·
HPC System Exascale
Automotive CPU

Gen3 GPP Family

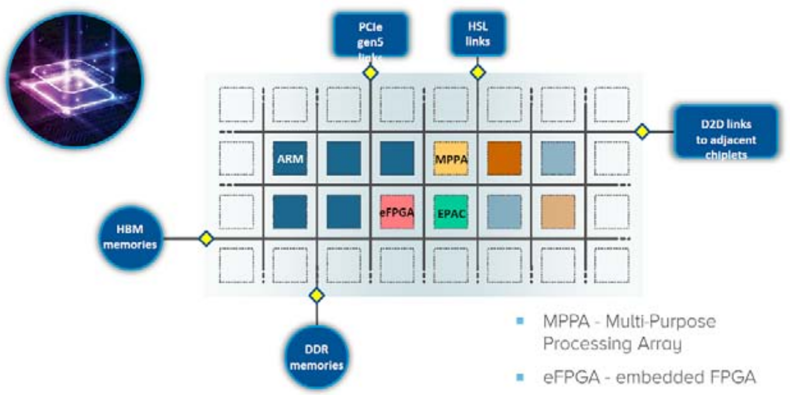
2024-...

arm

KALRAY

RISC-V

GPP AND COMMON ARCHITECTURE



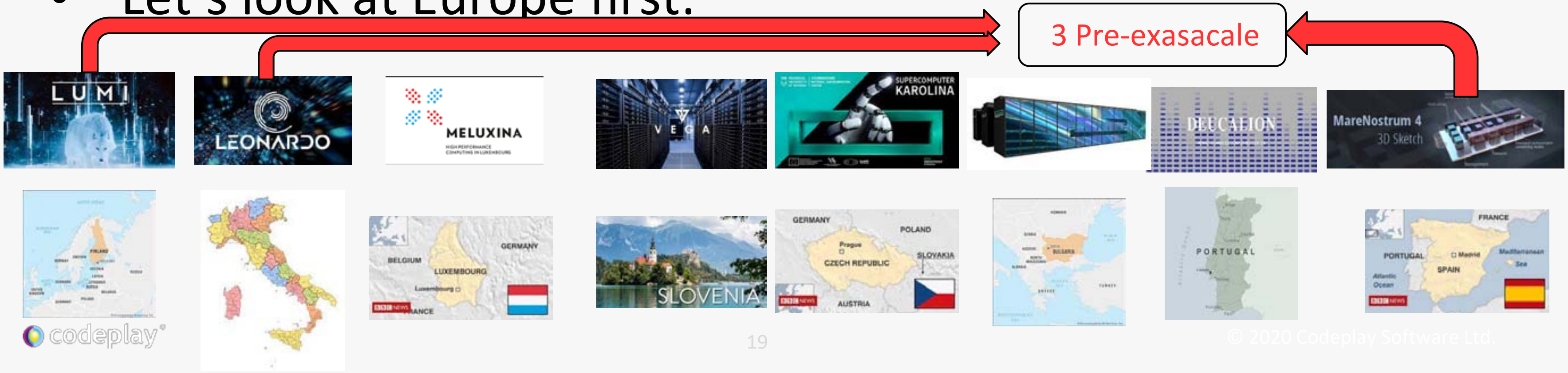
- MPPA - Multi-Purpose Processing Array
- eFPGA - embedded FPGA
- EPAC - EPI Accelerator

SYCL as a universal programming model for HPC

Starting with US National Labs

Across Europe, Asia are many Petascale and pre-exascale systems

- With many variety of CPUs GPUs FPGAs, custom devices
- Often with interconnected usage agreements
- Let’s look at Europe first:



3 Pre-exasacale

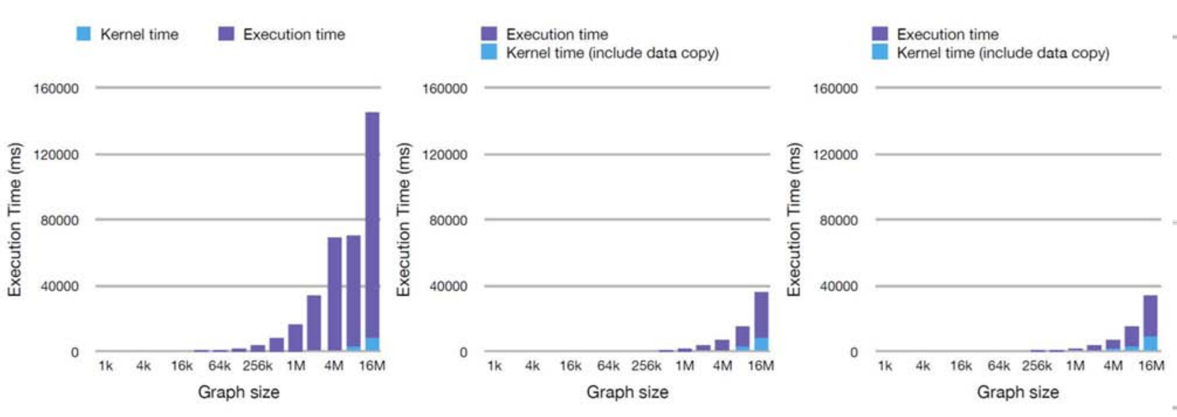
HPCAsia 2021: neoSYCL thanks to Hiroyuki Takizawa

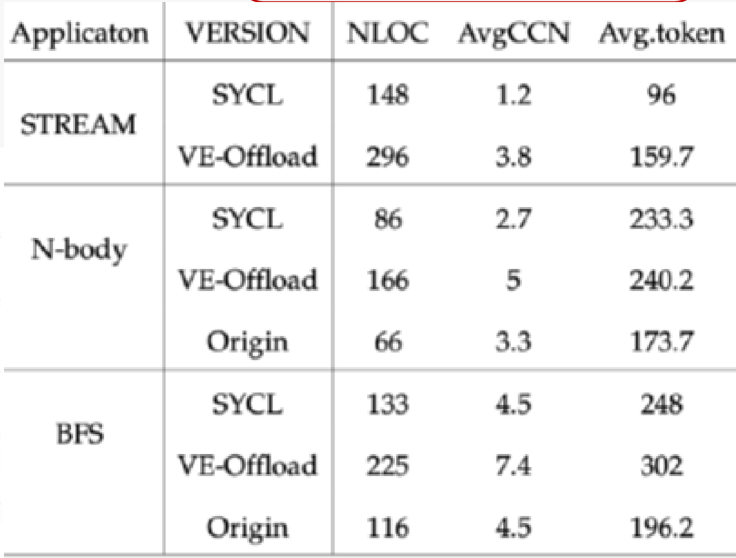
Open standard for offload programming = SYCL

BFS using Rodina Benchmark at HPC Asia 2021

No loss in performance between using SYCL and VEO

Programming with SYCL Leads to lower Code Complexity





Applicaton	VERSION	NLOC	AvgCCN	Avg.token
STREAM	SYCL	148	1.2	96
	VE-Offload	296	3.8	159.7
N-body	SYCL	86	2.7	233.3
	VE-Offload	166	5	240.2
	Origin	66	3.3	173.7
BFS	SYCL	133	4.5	248
	VE-Offload	225	7.4	302
	Origin	116	4.5	196.2

Final words

- SYCL can be a part of a standard programming model for all HPC including Europe/Asia/NA
    - HPC is now used in Embedded and Automotive
- SYCL is home grown EU, UK company led its development since 2012, now open standard with multiple company contributions, lots of European/Asia projects
    - Celerity from the University of Innsbruck and Salerno, CINECA Bologna, neoSYCL
- Moves with ISO C++, updates every 1.5-3 years
- Part of oneAPI
- Adapts to HPC hardware changes, moving towards safety critical
- Adapted by ECP for first Exascale computer in Aurora, now also in the Perlmutter, and we hope in European and Asia HPC

Enabling Industry Engagement

- SYCL working group values industry feedback
  - https://community.khronos.org/c/sycl
  - https://sycl.tech
- SYCL FAQ
  - https://www.khronos.org/blog/sycl-2020-what-do-you-need-to-know
- What features would you like in future SYCL versions?

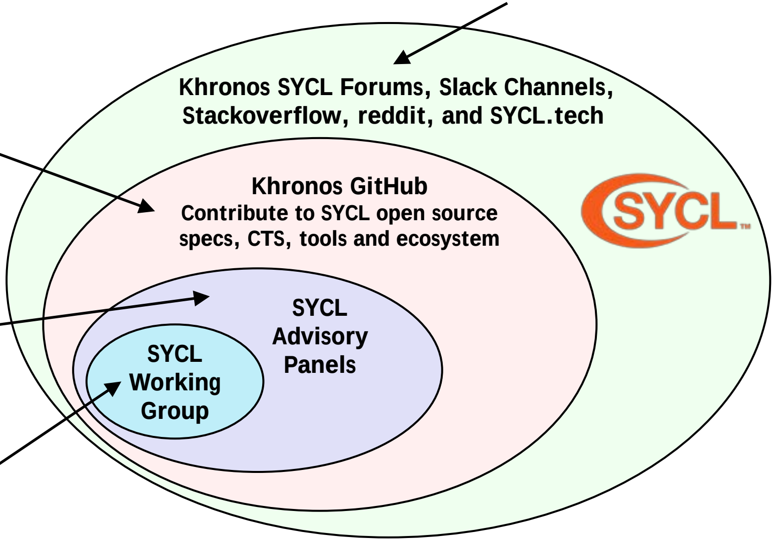
- Advisory Panel Chaired by Tom Deakin of U of Bristol
- SYCL Advisory Panel meeting here at IWOCL/SYCLCon
- Regular meetings to give feedback on roadmap and draft specifications

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/

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

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

Khronos members
https://www.khronos.org/members/
https://www.khronos.org/registry/SYCL/


Khronos SYCL Forums, Slack Channels, Stackoverflow, reddit, and SYCL.tech
Khronos GitHub
Contribute to SYCL open source specs, CTS, tools and ecosystem
SYCL Advisory Panels
SYCL Working Group
SYCL