codeplay

Enable AI & HPC to be Open, Safe and Accessible to All

Targeting Multi-Vendor Architectures with oneAPI and SYCL

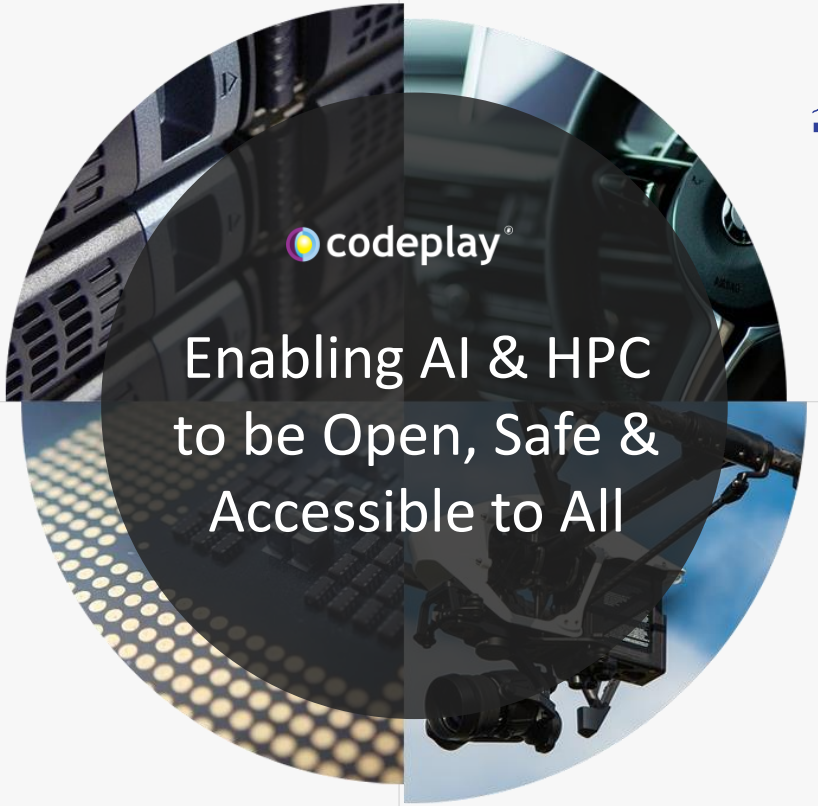
Rod Burns - Codeplay Software

Company

Leaders in enabling high-performance software solutions for new AI processing systems

Enabling the toughest processors with tools and middleware based on open standards

Established 2002 in Scotland, acquired by Intel in 2022 and now ~90 employees.



Collaborations

And many more!

Supported Solutions

oneAPI

An open, cross-industry, SYCL based, unified, multiarchitecture, multi-vendor programming model that delivers a common developer experience across accelerator architectures

Markets

High Performance Compute (HPC)
Automotive ADAS, IoT, Cloud Compute
Smartphones & Tablets
Medical & Industrial

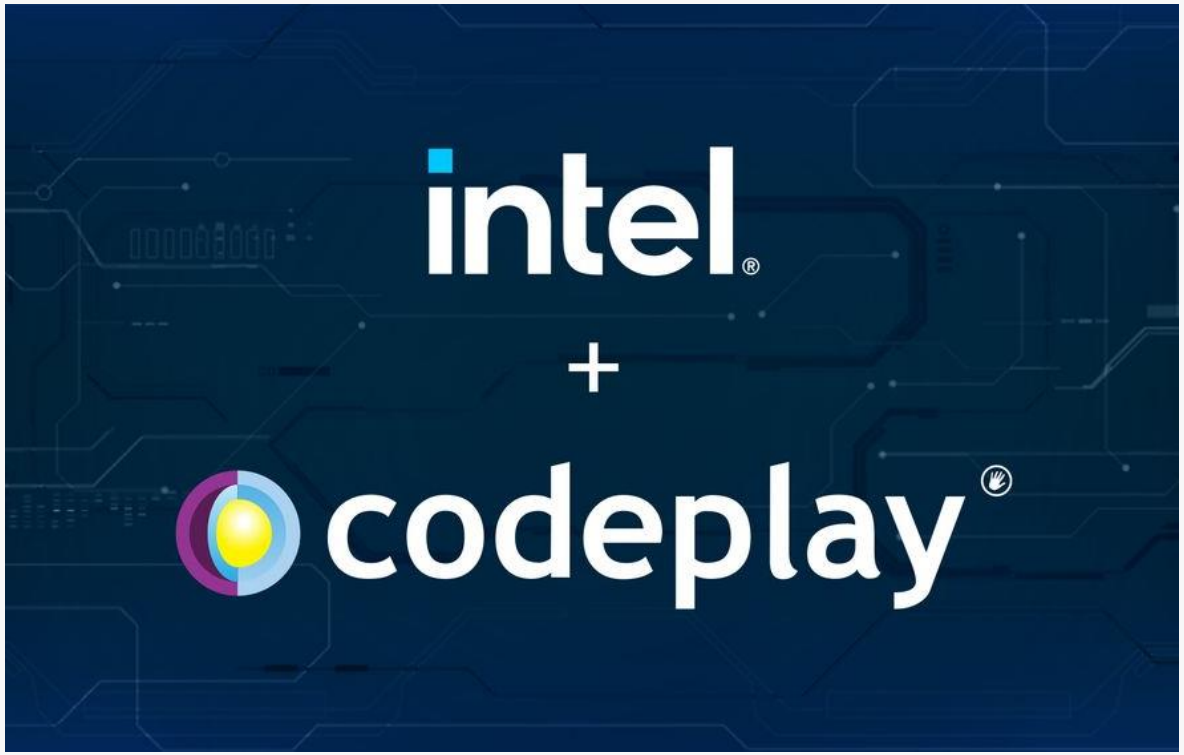
Technologies: Artificial Intelligence
Vision Processing
Machine Learning
Big Data Compute

Who we are

- After years of collaboration and contribution to open standards alongside **intel**, **Codeplay Software** is a subsidiary of **Intel** after an acquisition made this year.

- We will continue to operate as Codeplay Software and will work extensively with **all relevant industries** to **advance the SYCL ecosystem**, especially around **oneAPI**

- Codeplay is now working jointly with intel to further advance the **SYCL standard** and the **oneAPI** open ecosystem.



Open Standards Multi-Vendor Programming

- Open Standards Bring Freedom and Choice
- oneAPI and SYCL Deliver Performance Portability
- How to Use Familiar Hardware with Open Standards
- Get Started

codeplay

Enable AI & HPC to be Open, Safe and Accessible to All

Open Standards Bring Freedom and Choice

Say goodbye to proprietary lock-in

Impacts your

- Ability to use the best hardware, **regardless of vendor**
- Ability to **negotiate the best prices** for hardware

The remedy for lock-in is to **use products that conform to free, open standards**

https://en.wikipedia.org/wiki/Hazard_(golf)#/media/File:Road_hole_bunker.jpg

C++ and Open Standards

- C++ is back

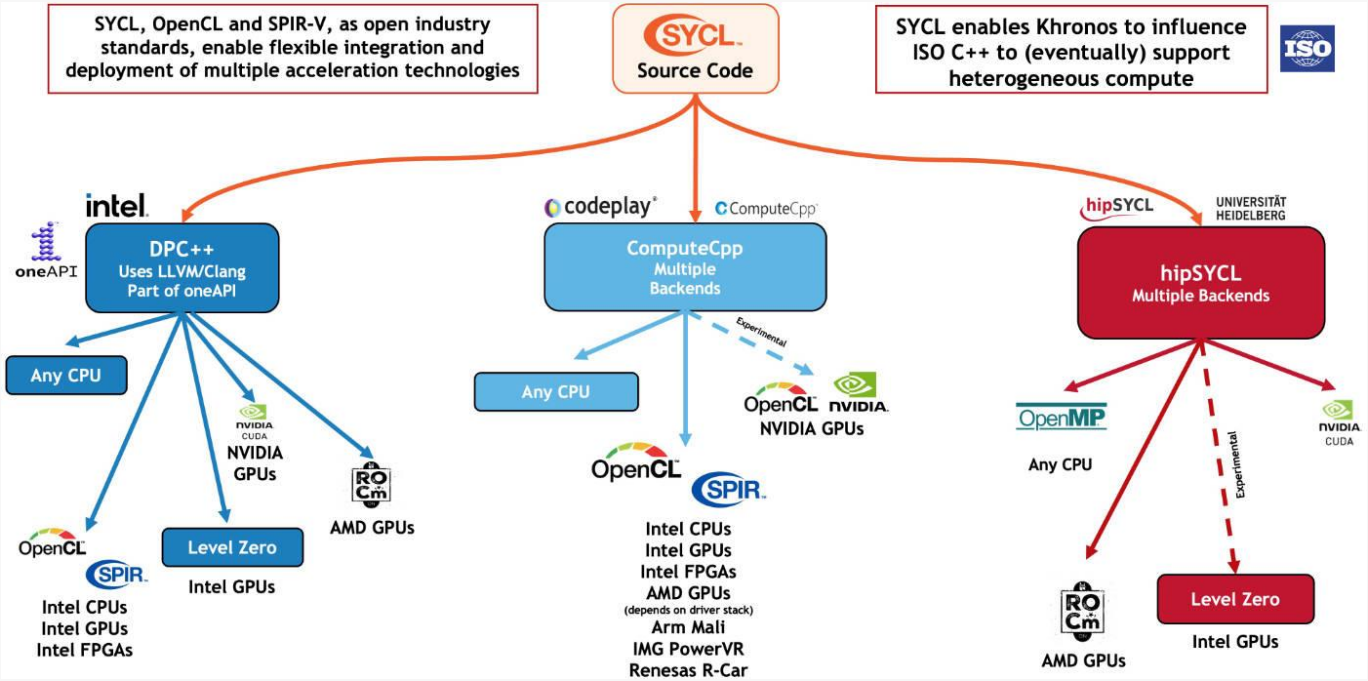
"fastest growth among the top 20 languages"

www.tiobe.com

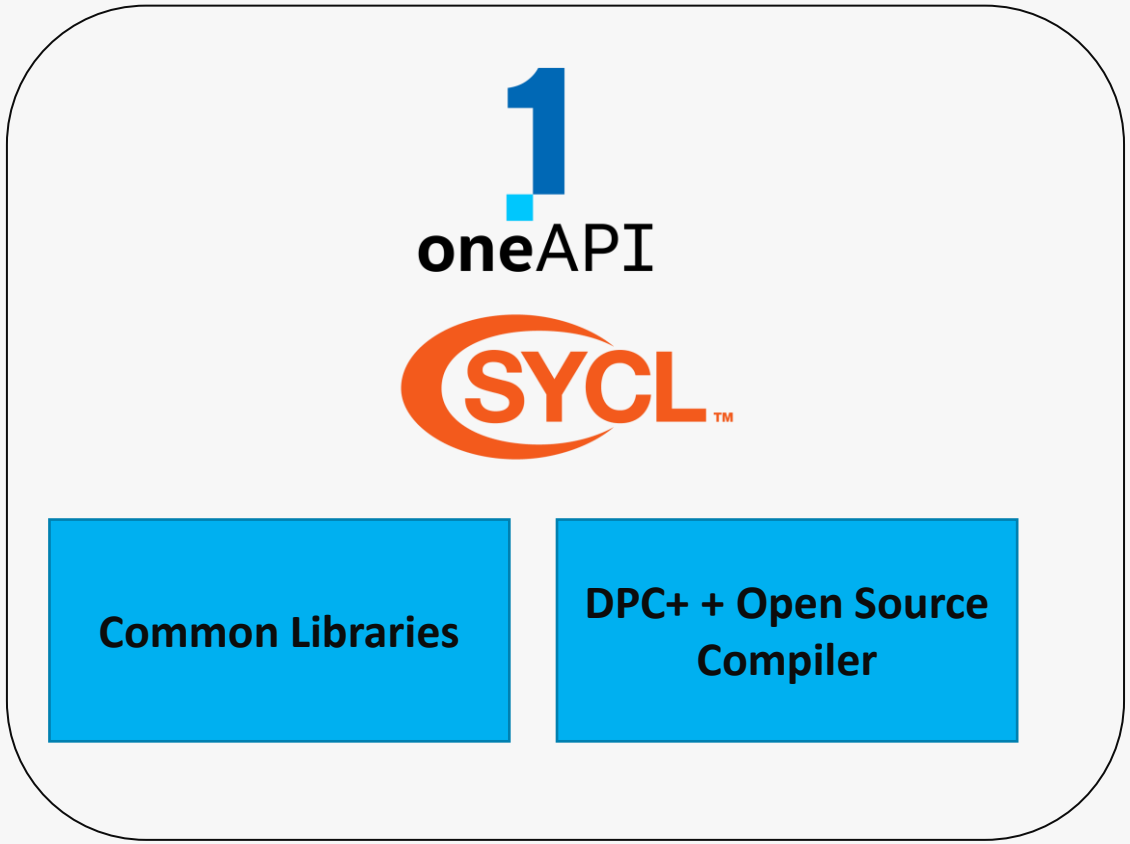
- A good choice for developing high performance computing applications
- Based on ISO ratified open standard specification

What is SYCL?

- Open standard specification
- Uses only standard C++
- Enables parallel execution
- Supports wide range of hardware



oneAPI and SYCL



- SYCL sits at the heart of oneAPI
- Open Source compiler project
- Common Libraries

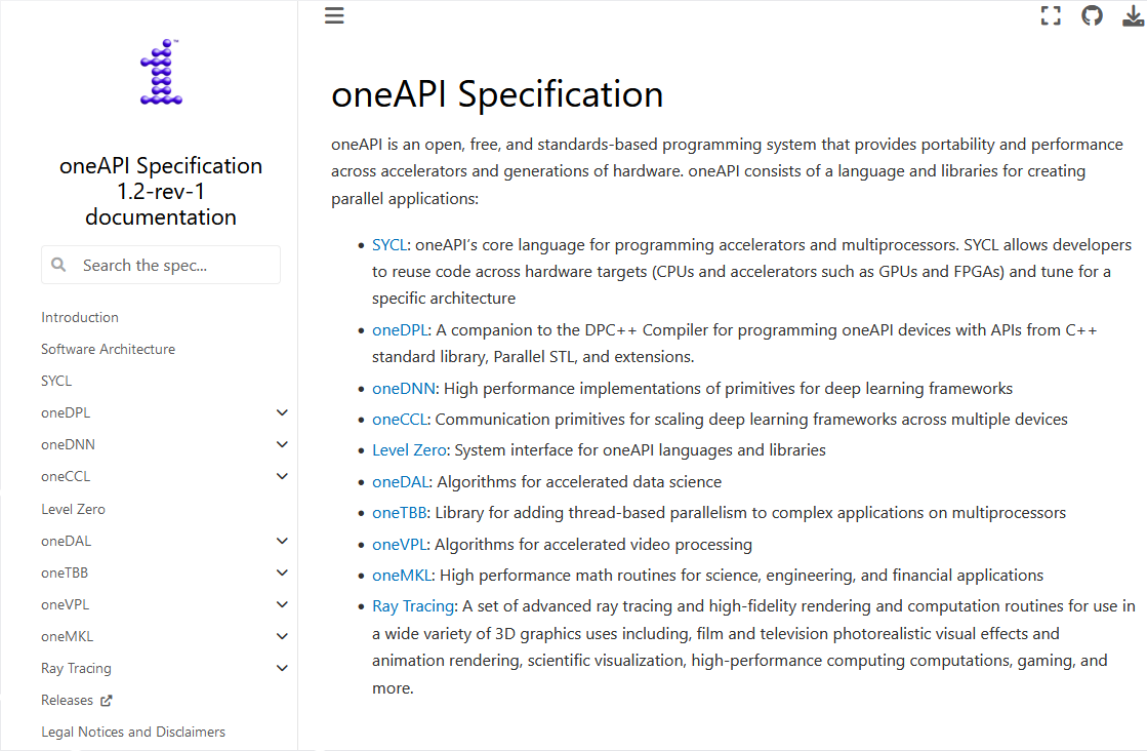
SYCL and oneAPI Bring Choice

- Deployed on some of the fastest supercomputers
- Software can run on current and next generation supercomputers

oneAPI Open Standard

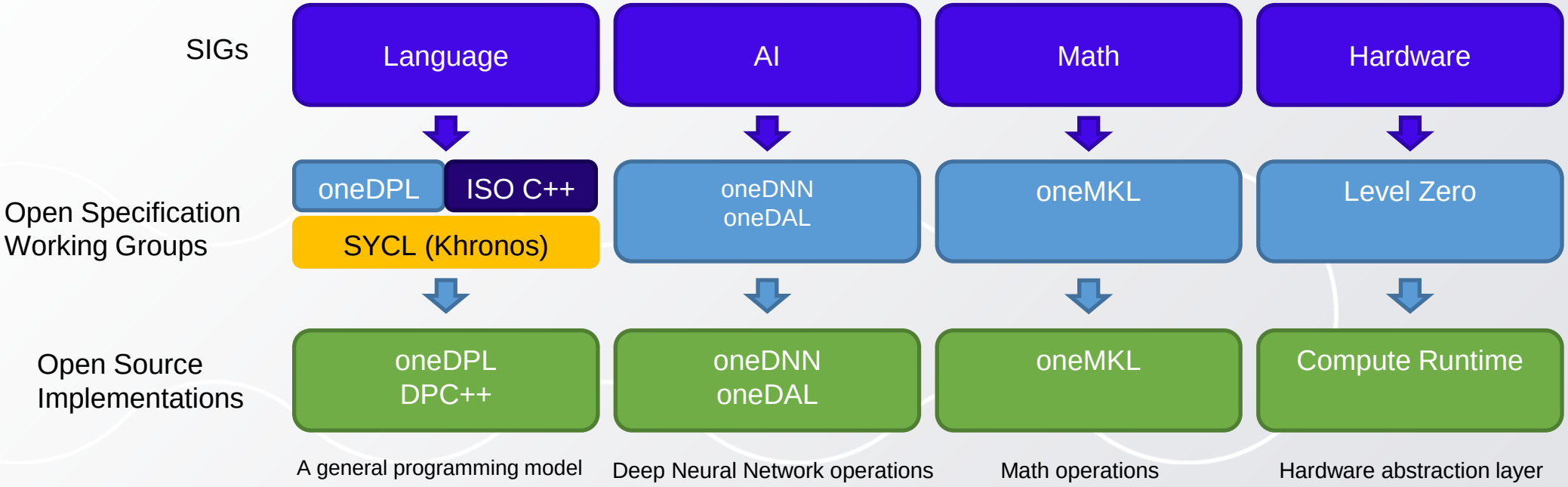
oneAPI Specification

- Dependency on SYCL standard
- Defines common libraries
- Open source implementations exist



Special Interest Groups (SIGs)

Special Interest Groups influence the specifications and implementations



Contribute to the oneAPI Community Forum

- Join and lead SIGs and Working Groups
- Lead technical discussions
- Submit proposals for features and changes
- Vote on proposals

Drive the future of programming for heterogeneous architectures

https://oneapi.io/community

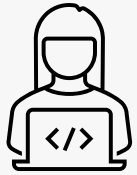
oneapi@codeplay.com

codeplay

Enable AI & HPC to be Open, Safe and Accessible to All

oneAPI and SYCL Deliver Performance Portability

Bringing **oneAPI** to NVIDIA and AMD GPUs



Write once.

Run anywhere.

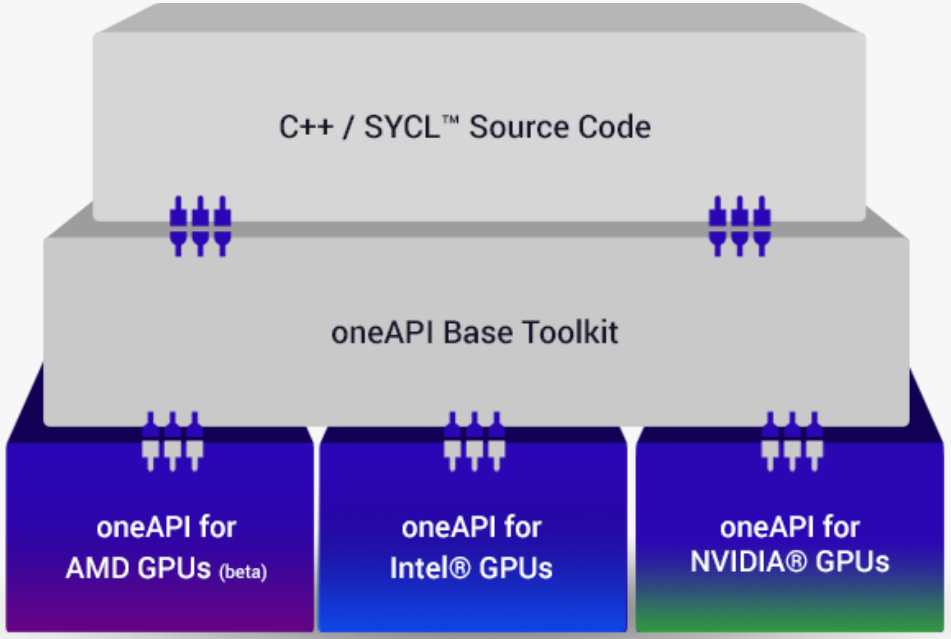
No compromises on performance.

Open, cross-industry collaboration on standards.

Write once.

Run anywhere.

Write code using **SYCL**, and then **run freely** across Intel, NVIDIA and AMD GPUs



No compromises on performance.

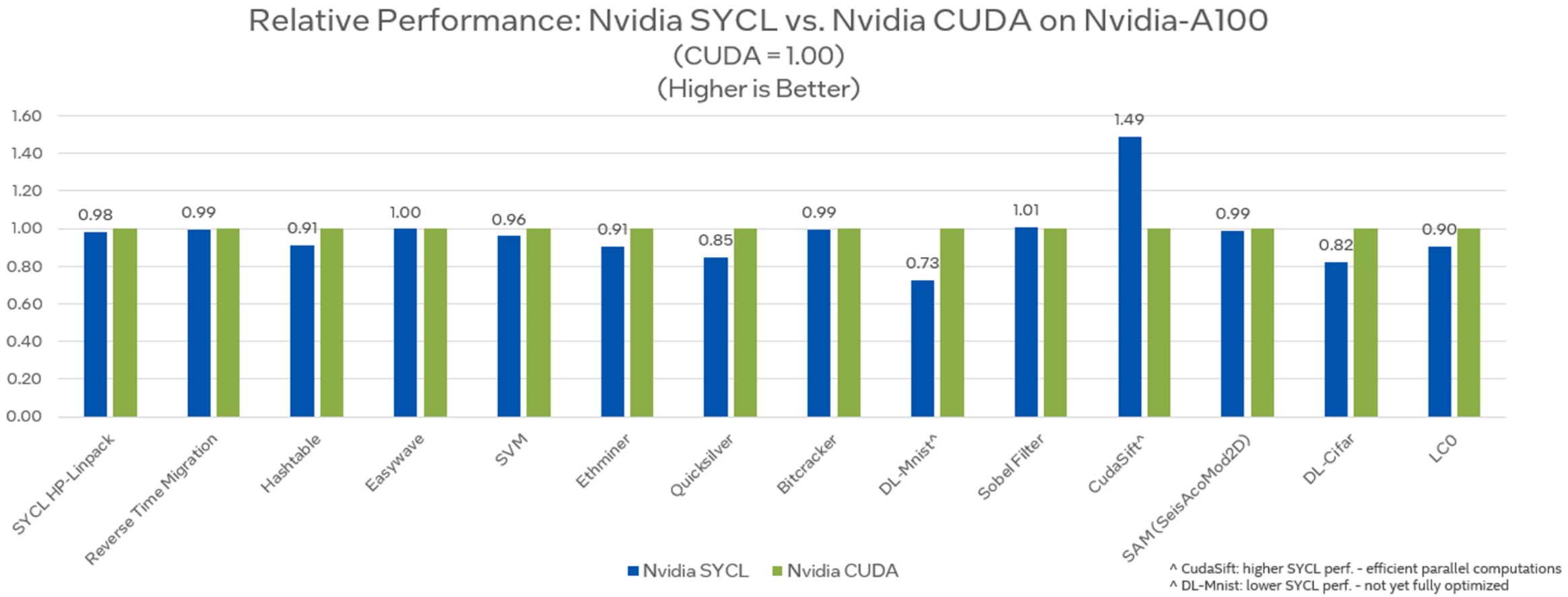
> SYCL is **highly performant** on Nvidia and AMD devices and **performs comparably** to native CUDA or HIP code for diverse workloads.

- Ruyman Reyes, CTO at Codeplay

Refer to https://codeplay.com/portal/blogs/2023/04/06/sycl-performance-for-nvidia-and-amd-gpus-matches-native-system-language for more information.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See configuration disclosure for details. No product or component can be absolutely secure. Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex. Your costs and results may vary.

On NVIDIA GPU – SYCL Provides Comparable Performance to CUDA

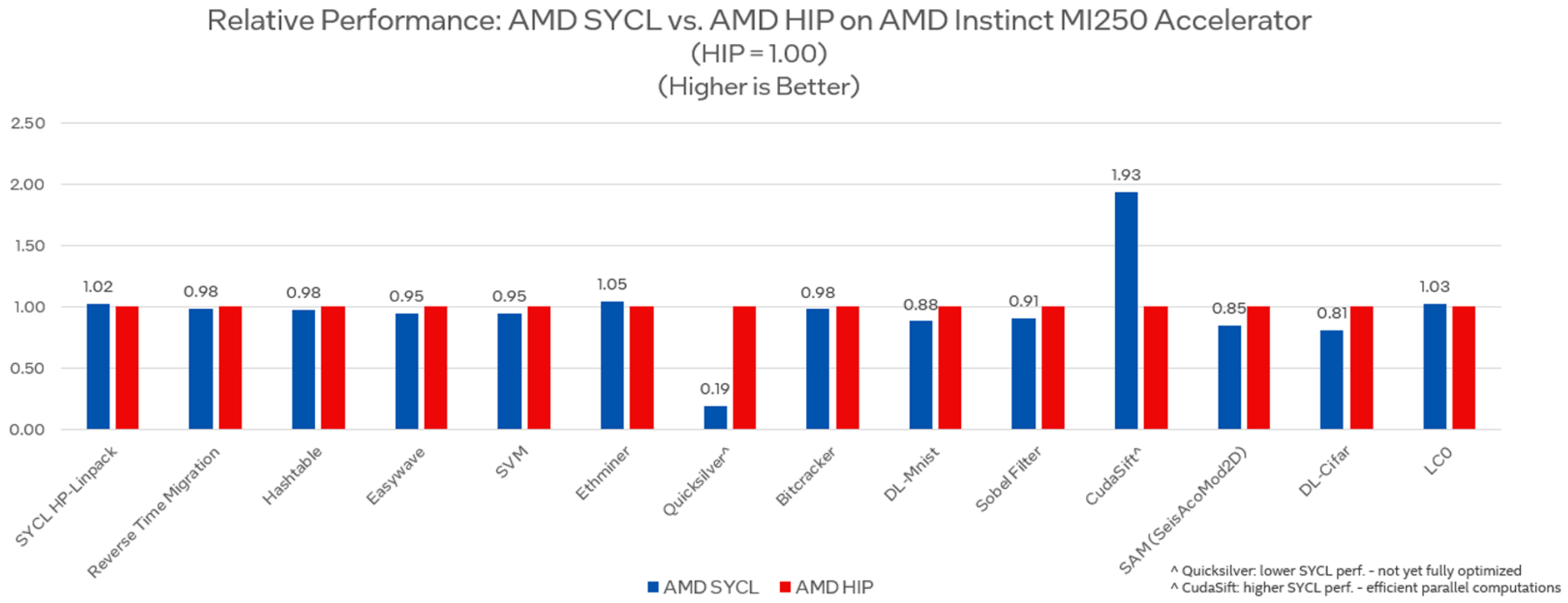


Testing Date: Performance results are **based on testing by Intel as of April 15, 2023** and may not reflect all publicly available updates.

Configuration Details and Workload Setup: Intel® Xeon® Platinum 8360Y CPU @ 2.4GHz, 2 socket, Hyper Thread On, Turbo On, 256GB Hynix DDR4-3200, ucode 0xd000363. GPU: Nvidia A100 PCIe 80GB GPU memory. Software: SYCL open source/CLANG 17.0.0, CUDA SDK 12.0 with NVIDIA-NVCC 12.0.76, cuMath 12.0, cuDNN 12.0, Ubuntu 22.04.1. SYCL open source/CLANG compiler switches: -fsycl-targets=nvptx64-nvidia-cuda, NVIDIA NVCC compiler switches: -O3 –gencode arch=compute_80, code=sm_80. Represented workloads with Intel optimizations.

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

On AMD GPU – SYCL Provides Comparable Performance to HIP



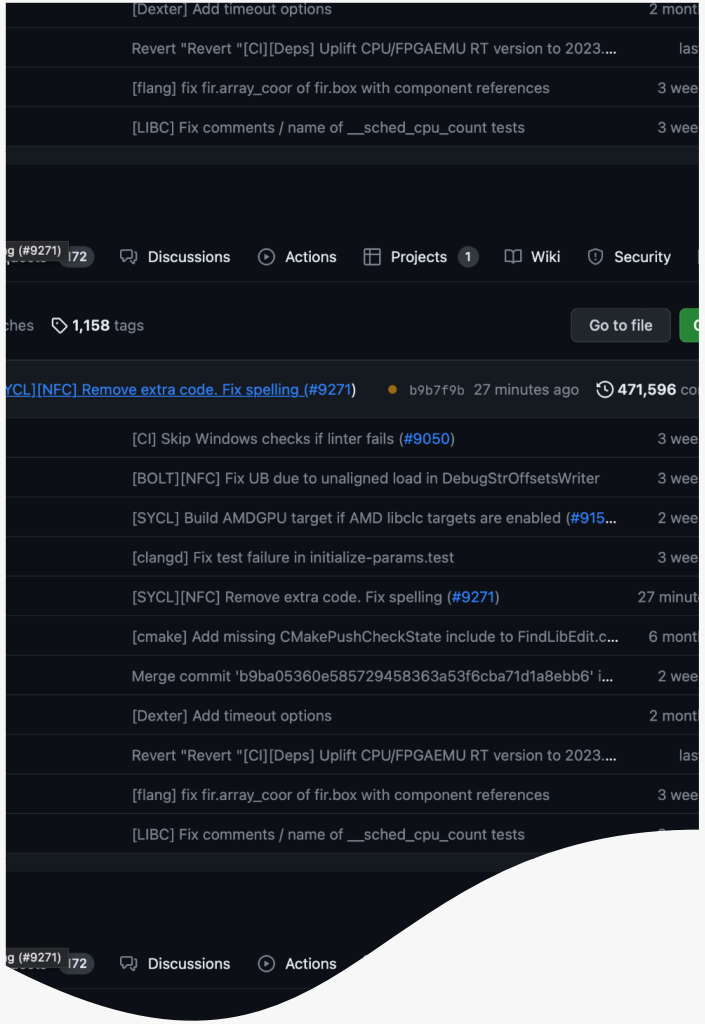
Testing Date: Performance results are **based on testing by Intel as of April 15, 2023** and may not reflect all publicly available updates.

Configuration Details and Workload Setup: AMD EPYC 7313 CPU @ 3.0GHz, 2 socket, AMD Simultaneous Multi-Threading Off, AMD Precision Boost Enabled, 512GB DDR4, ucode 0xa001144. GPU: AMD Instinct MI250 OAM, 128GB GPU memory. Software: SYCL open source/CLANG 17.0.0, AMD RoCm 5.3.0 with roc-5.3.0 22362, hipSolver 5.3.0, rocBLAS 5.3.0, Ubuntu 20.04.4. SYCL open source/CLANG compiler switches: -O3 -fsycl -fsycl-targets=amdgcn-amd-amdhsa -Xsycl-target-backend --offload-arch=gfx90a, AMD-ROCm compiler switches: -O3. Represented workloads with Intel optimizations.

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

Performance varies by use, configuration, and other factors. Learn more at www.Intel.com/PerformanceIndex. Your costs and results may vary.

Open, cross-industry collaboration on standards.



The code is entirely **open source**

Available as a **free plugin** on the Codeplay website

codeplay

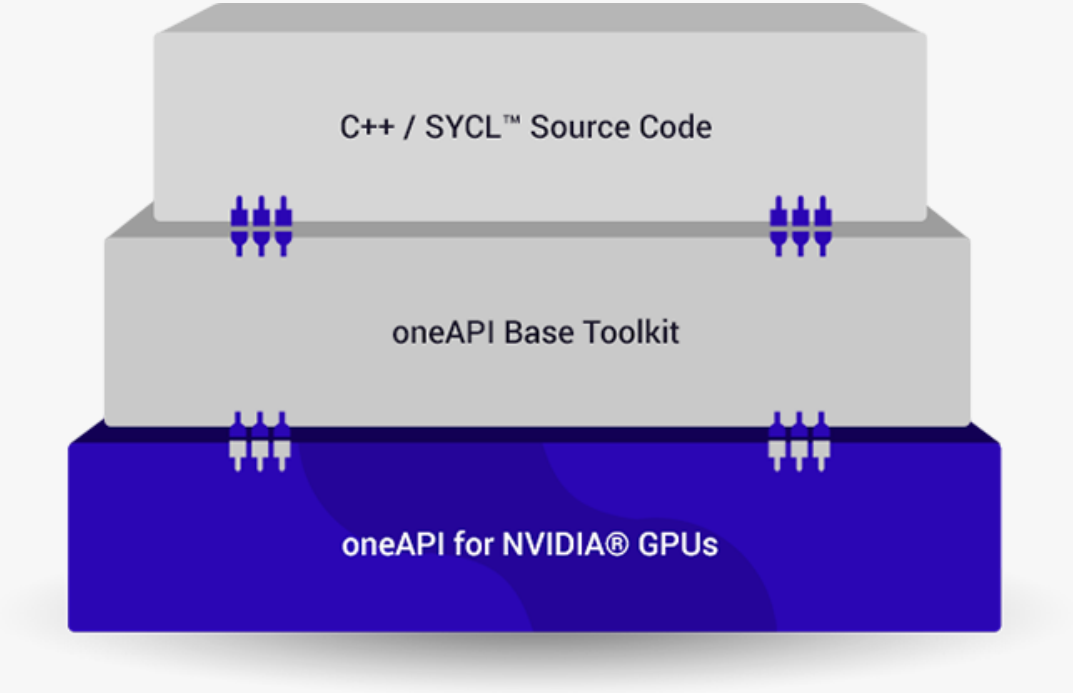
Enable AI & HPC to be Open, Safe and Accessible to All

Use Familiar Hardware with Open Standards

oneAPI for NVIDIA GPUs

Adds support for NVIDIA GPUs to the Intel oneAPI Base Toolkit.

Develop code using SYCL and run on NVIDIA GPUs.

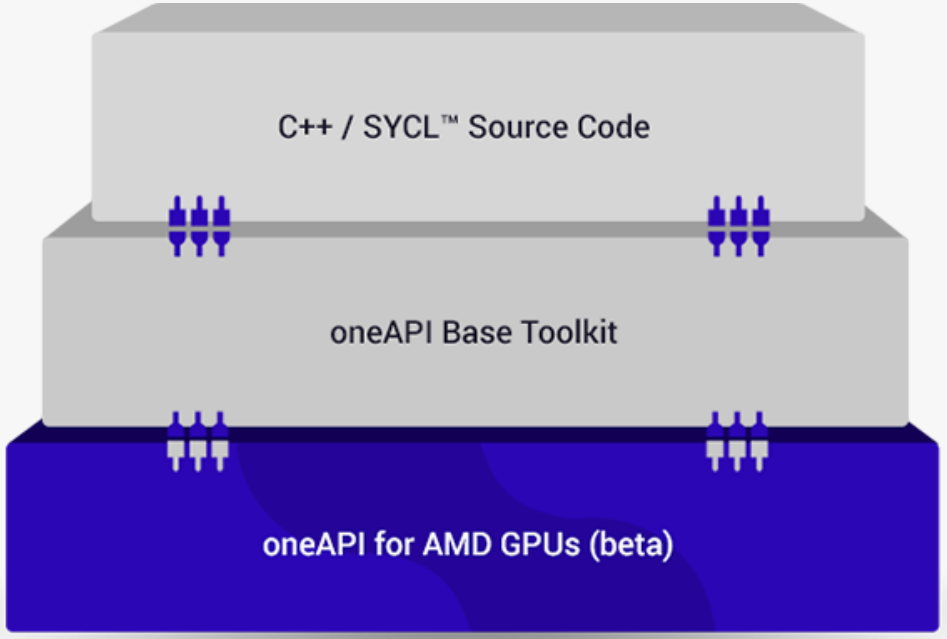


Download from developer.codeplay.com

oneAPI for AMD GPUs (beta)

Adds support for AMD GPUs to the Intel oneAPI Base Toolkit.

Develop code using SYCL and run on AMD GPUs.



Download from developer.codeplay.com

oneAPI for NVIDIA GPUs

- Ubuntu 22.04
- CUDA SDK >= 12.0
- GPUs with at least sm_50
- Primarily tested with A100 GPU

oneAPI for AMD GPUs

- Ubuntu 22.04
- HIP 5.4.1
- GPU Driver 6.1.0
- Primarily tested with MI50

Free release plugins
Entirely open source based

codeplay

Enable AI & HPC to be Open, Safe and Accessible to All

Get Started

Get Started

Get the pre-requisites

Install the CUDA or HIP development environment and drivers

Download the oneAPI Base Toolkit

Get the toolkit from Intel intel.com/developer

Download the oneAPI for Nvidia and AMD GPUs

Get the plugins from Codeplay developer.codeplay.com

Compiling

```
icpx -fsycl -fsycl-targets=nvptx64-nvidia-cuda sycl-app.cpp -o sycl-app
```

- Use the SYCL compiler
- Compile for Nvidia
- The source file
- The binary

Run the binary using this command

```
ONEAPI_DEVICE_SELECTOR=cuda:* SYCL_PI_TRACE=1 ./simple-sycl-app
```

- Tells the runtime to use Nvidia GPU
- Set the output trace to see warnings etc.

Multi-Target Compilation

Compiling a single binary for multiple targets

Compile for AMD

Compile for Nvidia

Compile for Intel

```
icpx -fsycl -fsycl-targets=amdgcn-amd-amdhsa,nvptx64-nvidia-cuda,spir64 \
-Xsycl-target-backend=amdgcn-amd-amdhsa --offload-arch=gfx1030 \
-Xsycl-target-backend=nvptx64-nvidia-cuda --offload-arch=sm_80 \
-o sycl-app sycl-app.cpp
```

Set NVIDIA architecture

Set AMD architecture

https://github.com/intel/llvm/blob/2ddbf8c3b7a5cb020b2df1776221c049948e9775/clang/include/clang/Basic/Cuda.h#L101

Multi-Target Compilation

Executing the binary on target hardware

Tells the runtime to use Nvidia GPU

```
ONEAPI_DEVICE_SELECTOR=cuda:* SYCL_PI_TRACE=1 ./simple-sycl-app
```

```
ONEAPI_DEVICE_SELECTOR=hip:* SYCL_PI_TRACE=1 ./simple-sycl-app
```

Tells the runtime to use AMD GPU

This can also be done through device selectors in code

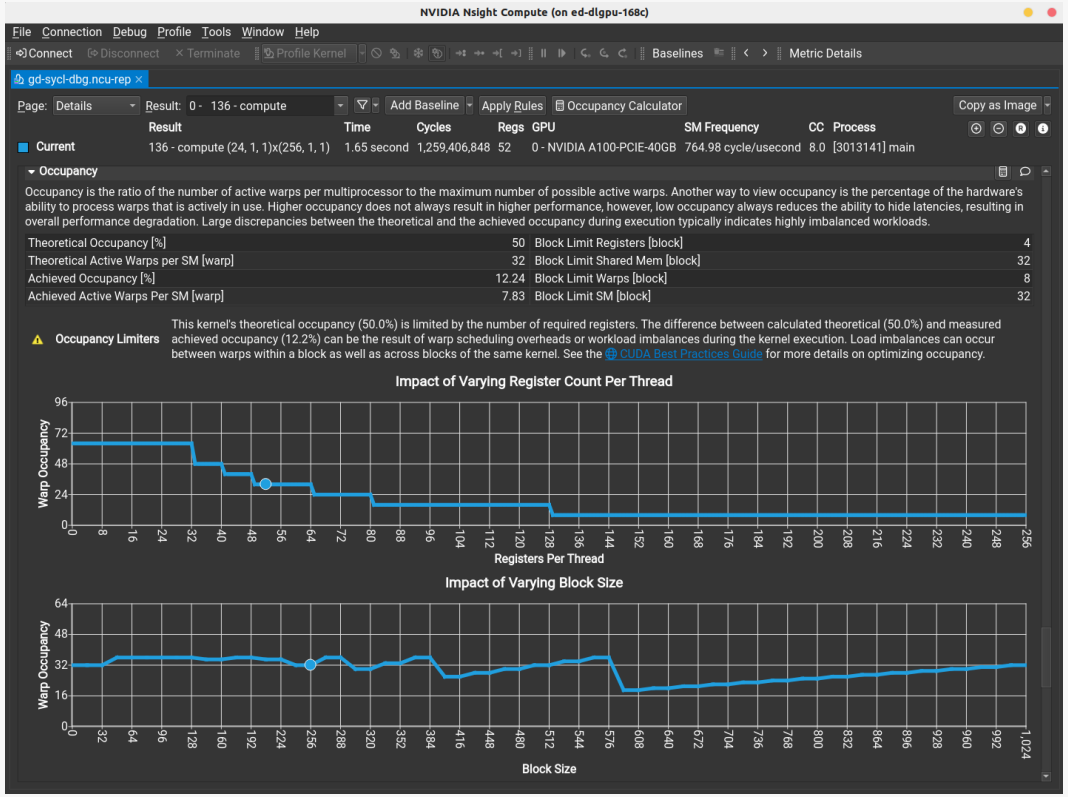
Debugging

- Use standard tooling for debugging
- gdb and the Nvidia specific CUDA-GDB
- VS Code integration

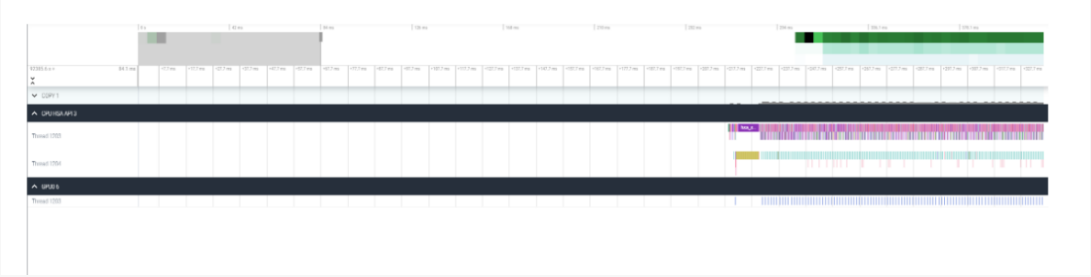
```
[Switching focus to CUDA kernel 1, grid 4, block (5,0,0), thread (32,0,0), device 0, sm 10, warp 0, lane 0]

Thread 1 "main" hit Breakpoint 1, main::{lambda(sycl::_V1::handler&)#5}::operator()(sycl::_V1::handler&) const
::{lambda(sycl::_V1::nd_item<1>)#1}::operator()(sycl::_V1::nd_item<1>) const (this=0x7fffa3fffb68, item=...) a
t main.cpp:115
115             float v = sycl::log(1+sycl::exp(-1*A_y_label[i]*xp));
(cuda-gdb) info cuda kernels
  Kernel Parent Dev Grid Status                      SMs Mask  GridDim  BlockDim Invocation

*      1      -   0    4 Active 0x00000000000000000555555555555 (24,1,1) (256,1,1) typeinfo name for main::{lam
bda(sycl::_V1::handler&)#5}::operator()(sycl::_V1::handler&) const::compute()
(cuda-gdb) list
110             for( int j = A_row_ptr[i]; j < A_row_ptr[i+1]; ++j){
111               xp += A_value[j] * x[A_col_index[j]];
112             }
113
114             // compute objective
115             float v = sycl::log(1+sycl::exp(-1*A_y_label[i]*xp));
116             auto atomic_obj_ref = atomic_ref<float,
117               memory_order::relaxed, memory_scope::device,
118               access::address_space::global_space> (total_obj_val[0]);
119             atomic_obj_ref.fetch_add(v);
(cuda-gdb) print i
$1 = 1312
(cuda-gdb) print xp
$2 = 0.0494509786
(cuda-gdb) next
118               access::address_space::global_space> (total_obj_val[0]);
(cuda-gdb) print v
$3 = 0.668727338
(cuda-gdb) continue
Continuing.
[Switching focus to CUDA kernel 1, grid 4, block (0,0,0), thread (0,0,0), device 0, sm 0, warp 3, lane 0]

Thread 1 "main" hit Breakpoint 1, main::{lambda(sycl::_V1::handler&)#5}::operator()(sycl::_V1::handler&) const
::{lambda(sycl::_V1::nd_item<1>)#1}::operator()(sycl::_V1::nd_item<1>) const (this=0x7fffa3fffb68, item=...) a
t main.cpp:115
115             float v = sycl::log(1+sycl::exp(-1*A_y_label[i]*xp));
(cuda-gdb) print xp
$4 = 0.0241964087
(cuda-gdb)
```

Profiling

- Use standard Nvidia profiling tooling



- Use standard AMD profiling tooling



codeplay

Enable AI & HPC to be Open, Safe and Accessible to All

Walkthrough

/home/dev

```
dev@demo $
```

Support for oneAPI for **NVIDIA GPUs**

Enterprise Support

Our highest level of support, for large teams.

Direct access to Codeplay's engineers and expertise via scheduled calls.

A custom support plan tailored to your requirements.

Priority Support

Suited to small teams and individuals.

Access to a ticketed support desk.

Accelerated response time for questions and requests.

Forum Support

A public forum moderated by Codeplay engineers.

Available for free.

Engage with the oneAPI community and our engineers.

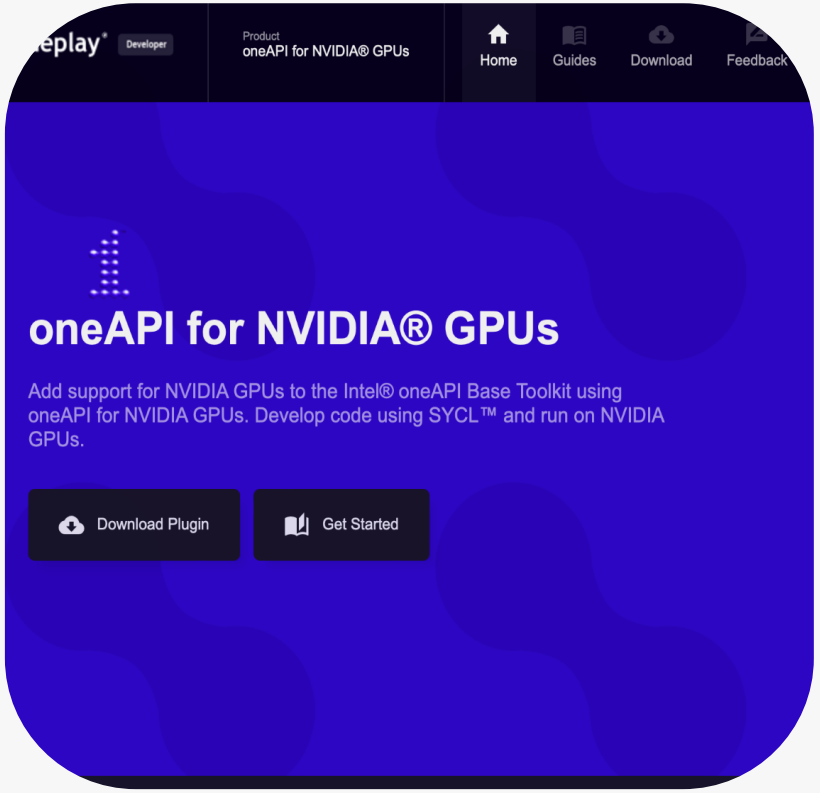
https://codeplay.com/company/contact/

https://support.codeplay.com

Summary

- Using open standard oneAPI and SYCL brings you choice
- You can influence the direction of oneAPI and SYCL
- Possible to achieve performance portability with oneAPI and SYCL
- You can use Codeplay's plugins to target Nvidia and AMD GPUs today

Try the Plugins



available for free at

https://developer.codeplay.com

Notices & Disclaimers

Performance varies by use, configuration and other factors.

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 technologies may require enabled hardware, software or service activation.

We're Hiring!

codeplay.com/careers/

codeplay®

Enable AI & HPC to be Open, Safe and Accessible to All

Questions

@codeplaysoft

info@codeplay.com

codeplay.com