OpenCL Working Group Update

Neil Trevett
Khronos President
OpenCL Working Group Chair
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d

IWOCL and SYCLcon 2022
10th International Workshops on OpenCL and SYCL
May 2022
Khronos Connects Software to Silicon

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

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

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

Proven multi-company governance and Intellectual Property Framework

Founded in 2000

~ 200 Members | ~ 40% US, 30% Europe, 30% Asia
Khronos Compute Acceleration Standards

- **Single source C++ programming with compute acceleration**
- **Graph-based vision and inferencing acceleration**

**Higher-level Languages and APIs**
Streamlined development and performance portability

**Lower-level Languages and APIs**
Direct Hardware Control

**GPU rendering + compute acceleration**

Intermediate Representation (IR) supporting parallel execution and graphics

Heterogeneous compute acceleration

**OpenCL Complements Vulkan**
Simpler programming model
Relatively lightweight run-time
More language flexibility, e.g., pointers
Rigorously defined numeric precision

- **CPU**
- **GPU**
- **FPGA**
- **DSP**
- **AI/Tensor HW**
- **Custom Hardware**
OpenCL Open-Source Project Momentum

# OpenCL-based GitHub Repos

Quadrupling in the last four years

This work is licensed under a Creative Commons Attribution 4.0 International License
OpenCL 3.0 Released September 2020

**Increased Ecosystem Flexibility**
All functionality beyond OpenCL 1.2 is queryable
Macros for optional OpenCL C language features
Widely adopted new extensions are integrated into new core specifications

**OpenCL C++ for OpenCL**
Open-source C++ for OpenCL front end compiler
Combines OpenCL C and C++17
Replaces OpenCL C++ language specification

**Unified Specification**
All versions of OpenCL in one specification
Easier maintenance, evolution and accessibility
Specification Source on Khronos GitHub for requests, feedback and bugs

**Moving Applications to OpenCL 3.0**
OpenCL 1.2 applications - no change
OpenCL 2.X applications - no code changes if all used functionality is present
Queries recommended for future portability

A Common Baseline for Ecosystem Evolution
OpenCL 3.0 Adoption

Product Conformance Status

https://www.khronos.org/conformance/adopters/conformant-products/opencl

OpenCL 3.0 Adopters

arm

Google

intel

NVIDIA

Qualcomm

codeplay

Imagination

Microsoft

QNX

VeriSilicon

OpenCL 3.0 Adopters

Shipping Conformant Implementations
Apps, Libraries and Engines using OpenCL

The industry’s most pervasive, cross-vendor, open standard for low-level heterogeneous parallel programming


Desktop Creative Apps
- Adobe
- otoy
- Modo
- Autodesk
- Chaos Group
- CyberLink
- REALFLOW
- LuxCoreRender
- Tenshil
- Xcode
- Unreal Engine

Parallel Languages
- OpenACC
- OpenCL
- SYCL
- OpenCL
- HIP

Machine Learning Libraries and Frameworks
- OpenVINO
- Intel CLDNN
- SYCL-DNN
- Caffe
- NNAPI
- VGG
- TensorFlow
- Metashape

Machine Learning Compilers
- Intel MKL
- NVIDIA CUDA
- AMD ROCm

Vision, Imaging and Video Libraries
- OpenCV
- Halide
- VisionCpp
- Gstreamer

Math and Physics Libraries
- Charmm
- OpenMM
- GROMACS
- ForceBalance
- OpenACC
- OpenCL

Molecular Modelling Libraries
- ForceBalance
- OpenMM
- GROMACS

Machine Learning Libraries
- TensorFlow
- PyTorch
- Keras
- Fast.ai

Vision, Imaging and Video Libraries
- OpenCV
- OpenCL
- Gstreamer

Math and Physics Libraries
- Charmm
- OpenMM
- GROMACS

Molecular Modelling Libraries
- ForceBalance
- OpenMM
- GROMACS

Linear Algebra Libraries
- SYCL-CLAS
- ViennaCL
- CLBlast

This work is licensed under a Creative Commons Attribution 4.0 International License
# OpenCL and Machine Learning

## Machine Learning Compilers

<table>
<thead>
<tr>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>tvm.ai</td>
</tr>
<tr>
<td>PlaidML</td>
</tr>
<tr>
<td>GLOW</td>
</tr>
<tr>
<td>Xilinx</td>
</tr>
</tbody>
</table>

## Import Formats

<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe, Keras, MXNet, ONNX</td>
</tr>
<tr>
<td>TensorFlow Graph, MXNet, PaddlePaddle, Keras, ONNX</td>
</tr>
<tr>
<td>PyTorch, ONNX</td>
</tr>
<tr>
<td>TensorFlow Graph, PyTorch, ONNX</td>
</tr>
</tbody>
</table>

## Front-end / IR

<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNVM / Relay IR</td>
</tr>
<tr>
<td>nGraph / Stripe IR</td>
</tr>
<tr>
<td>Glow Core / Glow IR</td>
</tr>
<tr>
<td>LLVM, TPU IR, XLA IR, TensorFlow Lite / NNAPI (inc. HW accel)</td>
</tr>
</tbody>
</table>

## Output

<table>
<thead>
<tr>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenCL, LLVM, CUDA, Metal</td>
</tr>
<tr>
<td>OpenCL, OpenCL, LLVM, CUDA</td>
</tr>
<tr>
<td>OpenCL, OpenCL, LLVM</td>
</tr>
<tr>
<td>LLVM, TPU IR, XLA IR, TensorFlow Lite / NNAPI (inc. HW accel)</td>
</tr>
</tbody>
</table>

## Common Steps

1. Import Trained Network Description
2. Graph-level optimizations e.g., node fusion, node lowering and memory tiling
3. Decompose to primitive instructions and emit programs for accelerated run-times

## Machine Learning Compilers and Frameworks using OpenCL Acceleration

<table>
<thead>
<tr>
<th>Libraries and Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibaba MNN</td>
</tr>
<tr>
<td>Arm Compute Library</td>
</tr>
<tr>
<td>Baidu PaddlePaddle/Paddle-Lite</td>
</tr>
<tr>
<td>Caffe</td>
</tr>
<tr>
<td>Intel cLDNN and OpenVINO</td>
</tr>
<tr>
<td>Google TensorFlow and NNAPI</td>
</tr>
<tr>
<td>SYCL-DNN</td>
</tr>
<tr>
<td>Synopsis MetaWare EV</td>
</tr>
<tr>
<td>Texas Instruments DL Library (TIDL)</td>
</tr>
<tr>
<td>VeriSilicon Acuity</td>
</tr>
<tr>
<td>Xiaomi Mace</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Embedded NN Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEVA Deep Neural Network (CDNN)</td>
</tr>
<tr>
<td>Cadence Xtensa</td>
</tr>
<tr>
<td>Neural Network Compiler (XNNC)</td>
</tr>
</tbody>
</table>

---

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

© The Khronos® Group Inc. 2022 - Page 8
C++ for OpenCL

Open-Source Compiler Front-end
Replaces the OpenCL C++ kernel language spec
Official release published in OpenCL-Docs repo

Enables full OpenCL C and most C++17 capabilities
OpenCL C code is valid and fully compatible
Enables gradual transition to C++ for existing apps

Supported in Clang since release 9.0
Generates SPIR-V 1.0 plus SPIR-V 1.2 where necessary
Online compilation via cl_ext_cxx_for_opencl extension

Check it out in Compiler Explorer

OpenCL Compiler Flow

Experimental Clang/LLVM native support for SPIR-V in Clang 14.0.0
SPIR-V enables a rich ecosystem of languages and compilers to target low-level APIs such as Vulkan and OpenCL, including deployment flexibility: e.g., running OpenCL kernels on Vulkan.
# API Layering

**Layers Over**

<table>
<thead>
<tr>
<th>Layers Over</th>
<th>Vulkan</th>
<th>OpenGL</th>
<th>OpenCL</th>
<th>OpenGL ES</th>
<th>DX12</th>
<th>DX9-11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulkan</td>
<td></td>
<td>Zink</td>
<td>clspv</td>
<td>GLOVE Angle</td>
<td>vkd3d-Proton</td>
<td>DXVK WineD3D</td>
</tr>
<tr>
<td>OpenGL</td>
<td>gfx-rs</td>
<td></td>
<td>clvk</td>
<td>Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WineD3D</td>
</tr>
<tr>
<td>DX12</td>
<td>Dozen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Microsoft D3D11On12</td>
</tr>
<tr>
<td></td>
<td>gfx-rs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘GLOn12’</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DX9-11</td>
<td>gfx-rs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Angle</td>
</tr>
<tr>
<td></td>
<td>Ashes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MoltenGL Angle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>gfx-rs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COLUMNS Benefit ISVs by making an API available everywhere**

**ROWS Benefit Platforms by adding APIs**

Enabled by growing robustness of open-source compiler ecosystem using SPIR-V
Layered OpenCL Implementations

**clspv + clvk**
- clspv - Google’s open-source OpenCL kernel to Vulkan SPIR-V compiler
- Tracks top-of-tree LLVM and Clang - not a fork
- Clvk - prototype open-source OpenCL to Vulkan run-time API translator
- Used by shipping apps and engines on Android e.g., Adobe Premiere Rush video editor - 200K lines of OpenCL C kernel code

**OpenCLon12**
- Microsoft and COLLABORA
- GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)
- Leverages Clang/LLVM AND MESA
- OpenCLon12 - OpenGL 3.3 over DX12 is already conformant

---

OpenCL C or C++ for OpenCL Kernel Sources

<table>
<thead>
<tr>
<th>Clang+Clspv Compiler</th>
<th>OpenCL Application Host Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>clspv + clvk</strong></td>
<td>OpenCL over Vulkan</td>
</tr>
</tbody>
</table>

Vulkan SPIR-V

Vulkan Runtime

OpenCL C or C++ for OpenCL Kernel Sources

<table>
<thead>
<tr>
<th>Clang+LLVM+SPIR-V LLVM</th>
<th>OpenCL SPIR-V</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mesa SPIR-V to DXIL</strong></td>
<td></td>
</tr>
</tbody>
</table>

Translates through MESA’s NIR Intermediate Representation

DX12 Runtime

---

OpenCLon12 Run-time API Translator

---

Microsoft and COLLABORA

GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)

Leverages Clang/LLVM AND MESA

OpenCLon12 - OpenGL 3.3 over DX12 is already conformant

---

OpenCL

OpenCL over DX12

---

Google

clspv - Google’s open-source OpenCL kernel to Vulkan SPIR-V compiler

Tracks top-of-tree LLVM and Clang - not a fork

Clvk - prototype open-source OpenCL to Vulkan run-time API translator

Used by shipping apps and engines on Android e.g., Adobe Premiere Rush video editor - 200K lines of OpenCL C kernel code

---

Vulkan

OpenCL over Vulkan

---

Microsoft

OpenCL on any DX12 PC and Cloud instance (x86 or Arm)

Leverages Clang/LLVM AND MESA

OpenCL on DX12

---

Colabora

GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)

Leverages Clang/LLVM AND MESA

OpenCL on DX12
Regular OpenCL Specification Releases

OpenCL 3.0.11 shipped on May 6th, 2022
Continues the regular release cadence for new functionality and bug fixes

New OpenCL extensions shipped since IWOCL 2021
Subgroup rotate extension for efficient data exchange among work-items
Workgroup Uniform Arithmetic for new work-group scan and reduction operators
Command Buffers Record and Replay (provisional)
  Asynchronous DMA
  Expect Assume Hints
  Enhanced subgroup functionality
  Extended bit-level operations
Universally unique device identifier query
Enhanced queries for platform and device versions
SPIR-V support for C++ linkage types
Integer Dot Product for Faster Neural Network Inferencing
External memory objects and semaphores for external sharing and Interop (provisional)
OpenCL Added to GPUInfo.org

The online GPUinfo.org database is populated using the OpenCL Hardware Capability Viewer application. Available for Windows, Linux and Android.

Reads and displays OpenCL information and uploads to the database.

Please download and run to help populate the database!

The @openclapi #opencl hardware database is also off to a good start with more than 140,000 page hits per month :)

10:36 AM - Apr 28, 2022 - Twitter Web App
OpenCL SDK Upgrades

Open-source OpenCL SDK includes all components to develop OpenCL applications

- OpenCL Headers (include/api)
- OpenCL C++ bindings (include/cpp)
- OpenCL Utility Libraries (include/utils)
- Build system and CI

Documentation and Sample Code

- OpenCL Guide
- Code samples (samples/)
- Documentation (docs/)

Loader and Layers

SDK and Layers Tutorial

Khronos funds SDK upgrades

Community contributions also welcome!

Spring 2022 SDK Updates

More details in the SDK Blog

Enhanced Cmake-based build system
- Subprojects and components
- Binary releases
- Tagged SDK versions

Enhanced SDK documentation
- In OpenCL Guide

OpenCL 3.0 Samples
- C, C++, Python and Ruby

Utility Libraries
- For loading kernel source and binary files

What’s Coming!

Upstream to Kitware’s FindOpenCL.cmake
- Enhances OpenCL:: namespace

Packaging and Distribution Support
- Build packages from the SDK
- Package newer versions of OpenCL
- Ease cross-platform installation, including PPAs

Enhanced SDK Validation Layers
- Object lifetime, Input parameters, SPIR-V
OpenCL Roadmap

OpenCL Extension Pipeline
Provisional and Vendor Extensions - Candidates for Ratification
We are listening to your input!

Support C++ for OpenCL
External Memory Export (in provisional release)
Command Buffer Record/Replay (in provisional release)
Unified Shared Memory
Floating Point Atomics
Required Subgroup Size
Generalized Image from buffer

Image Tiling Controls
YUV Multi-planar Images
Cross-workgroup Barriers
Cooperative Matrices
Timeline Semaphores
32 and 64-length vectors
Indirect Dispatch

OpenCL Developer Survey 2022
Survey is still open!
Please let us know what functionality is important to you!
OpenCL Advisory Panel

Working Group

Khronos Members
Any company can join. Membership Fee. Covered by NDA and IP Framework

Advisory Panel

Panel Members
Invited industry experts. $0 Cost. Covered by NDA and IP Framework

Shared Email list and Repository

Hosted by Khronos. Under Khronos NDA

Working Group makes decisions on standards evolution

Working groups can share draft specifications and accept detailed design contributions as Panel Members are covered by IP Framework

Chaired by Máté Ferenc Nagy-Egri at StreamHPC
Regular meetings to give feedback on roadmap and draft specifications

Please reach out to opencl-chair@lists.khronos.org if you wish to apply
Developers - Please Give Us Feedback!

• Give us your feedback on the OpenCL spec GitHub
  - What could be added to the OpenCL ecosystem to make you more productive?
  - What API and Language features do you most need?
  - https://github.com/KhronosGroup/OpenCL-Docs

• Please download and run the GPUinfo OpenCL Hardware Capability Viewer
  - https://opencl.gpuinfo.org/download.php

• Consider applying to join the OpenCL Advisory Panel!
  - Email opencl-chair@lists.khronos.org

• Take the 2022 OpenCL Developer Survey!
  - https://www.surveymonkey.com/r/J2BCQJN
OpenCL Resources

• OpenCL Home Page
  - https://www.khronos.org/opencl/

• OpenCL Registry for OpenCL core and extension specifications
  - https://www.khronos.org/registry/OpenCL/

• C++ for OpenCL Documentation
  - https://github.com/KhronosGroup/Khronosdotorg/blob/master/api/opencl/assets/CXX_for_OpenCL.pdf

• OpenCL SDK
  - https://github.com/KhronosGroup/OpenCL-SDK

• OpenCL Guide
  - https://github.com/KhronosGroup/OpenCL-Guide

• OpenCL Specification Source
  - https://github.com/KhronosGroup/OpenCL-Docs

• OpenCL Conformant Products
  - https://www.khronos.org/conformance/adopters/conformant-products/opencl

• GPUinfo.org Hardware Database
  - https://www.gpuinfo.org/

• Layered OpenCL implementations - clspv/clvk and OpenCLon12
  - https://github.com/google/clspv
  - https://github.com/kpet/clvk
  - https://github.com/microsoft/OpenCLon12