KHRONOS CROSS-PLATFORM STANDARDS UPDATE
VULKAN, ANARI, OPENXR, GLTF AND OPENCL

Neil Trevett, GTC, March 2020
VP NVIDIA, Khronos President
AGENDA

Latest updates on key Khronos open standards
That are of most interest to the GTC audience
Vulkan, ANARI, SPIR-V, OpenXR, glTF and OpenCL

On Khronos format slides

NVIDIA and Khronos Standards
How NVIDIA is supporting and deploying
Khronos Open Standards

On NVIDIA format slides
Khronos Connects Software to Silicon

Open interoperability standards to enable software to effectively harness the power of multiprocessors and accelerator silicon

3D graphics, XR, parallel programming, vision acceleration and machine learning

Non-profit, member-driven standards-defining industry consortium

Open to any interested company

All Khronos standards are royalty-free

Well-defined IP Framework protects participant’s intellectual property

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

3D Graphics
Desktop, Mobile, Web
Embedded and Safety Critical

3D Assets
Authoring and Delivery

Portable XR
Augmented and Virtual Reality

Parallel Computation
Vision, Inferencing, Machine Learning

Guidelines for creating APIs to streamline system safety certification

NVIDIA is fully committed to continuing to support OpenGL
Exploring expanded interop with Vulkan
New functionality is primarily Vulkan-focused
Pervasive Vulkan

Desktop and Mobile GPUs

Platforms

- Desktop
- Android (Android 7.0+)
  (Vulkan 1.1 required on Android Q)
- Apple (via porting layers)
- Media Players
- Consoles
- Virtual Reality
- Cloud Services
- Game Streaming
- Embedded

Engines

- CIDER
- XENKO
- GODOT
- CRYENGINE
- unity
- source
- UNREAL
- Crotoam Serious Engine
- NetEase Games

Note: The version of Vulkan available will depend on platform and vendor

http://vulkan.gpuinfo.org/
Vulkan AAA Content
Vulkan 1.1 Extensions
- Maintenance updates plus additional functionality
- Timeline semaphores
- DX/HLSL compatibility
- Bindless resources
- Reduced precision arithmetic
- Formal memory model
- Buffer references
- SPIR-V 1.5

Roadmap Discussions
- Ray Tracing
- Variable Rate Shading
- Accelerated Video Encode/Decode
- Machine Learning Primitives
- Mesh Shaders

January 2020

DLSS 2.0 smart scaling can be used with Vulkan applications
Does not need per application training
# Vulkan Ray Tracing

Set of Extensions to Vulkan, GLSL and SPIR-V
Seamlessly integrates ray tracing into Vulkan 1.X

Familiar Ray Tracing Pipeline Architecture
Straightforward porting between Vulkan Ray Tracing and DXR
- including re-use of ray tracing shaders written in HLSL

<table>
<thead>
<tr>
<th></th>
<th>Vulkan Ray Tracing</th>
<th>DX12 / DXR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ray Tracing Pipelines</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ray Queries</td>
<td>Optional</td>
<td>DXR 1.1 Inline raytracing</td>
</tr>
<tr>
<td>Language for Ray Tracing Shaders</td>
<td>GLSL or HLSL</td>
<td>HLSL</td>
</tr>
<tr>
<td>Pipeline Libraries</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Build Acceleration Structure on Host</td>
<td>Optional</td>
<td>No</td>
</tr>
<tr>
<td>Deferred Host Operations</td>
<td>Optional</td>
<td>No</td>
</tr>
<tr>
<td>Capture/Replay Support for Tools (e.g. RenderDoc)</td>
<td>Optional</td>
<td>No</td>
</tr>
</tbody>
</table>

The industry's first open, cross-vendor, cross-platform standard for ray tracing acceleration
Can be accelerated on existing GPU compute and dedicated ray tracing cores

Extensions are Provisional
Launched 17th March 2020
Open to developer feedback before finalization
https://khr.io/vkrayprovfeedback

Straightforward port from NVIDIA VKRay vendor extension to KHR extensions
Shipping beta drivers today
Example code coming soon
https://www.khronos.org/registry/vulkan/
HLSL and Vulkan with DXC
Microsoft’s DXC HLSL compiler was open sourced in Jan 2017
Google and others have added SPIR-V code generation to DXC
with Microsoft’s knowledge and approval
Vulkan developers can now choose between GLSL and HLSL!

Vulkan Ray Tracing includes GLSL and SPIR-V Extensions
Enabling compiled GLSL/SPIR-V shaders to operate in a
Ray Tracing Pipeline - similar to HLSL features used in Direct3D’s DXR

HLSL for Vulkan Ray Tracing
NVIDIA added code generation to DXC to generate SPIR-V for the
NVIDIA VKRay ray tracing vendor extension from HLSL
Vulkan Ray Tracing Extensions supported in HLSL soon

Developers can port HLSL shaders
with minimal changes between Vulkan Ray Tracing and DXR
NVIDIA AND VULKAN

NVIDIA deeply engaged in Vulkan and driving extend/consolidate cycle

- NVIDIA shipped Vulkan 1.0, 1.1 and 1.2 on day of spec releases
- Shipped beta Vulkan Ray Tracing extensions on day of spec release
- Increased Vulkan support in NSIGHT 2020.2 development tool - close parity now with DX12

NVIDIA chairing multiple Vulkan initiatives at Khronos

- Ray Tracing: contributed NVIDIA VKRay to catalyze Vulkan Ray Tracing KHR extensions
- Vulkan Portability: bringing layered Vulkan to Apple, WebGPU, silicon without native drivers etc.
- Machine Learning: low-level inferencing primitives
## Open Source Layering Projects

**Fighting Platform Fragmentation**

<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 clvk</td>
<td>GLOVE Angle</td>
<td>vkd3d</td>
<td>DXVK WineD3D</td>
</tr>
<tr>
<td>OpenGL</td>
<td>gfx-rs Ashes</td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
<td>WineD3D</td>
</tr>
<tr>
<td>DX12</td>
<td>gfx-rs</td>
<td>Microsoft ‘GLOn12’</td>
<td>Microsoft ‘CLOn12’</td>
<td></td>
<td>Microsoft D3D11On12</td>
<td></td>
</tr>
<tr>
<td>DX9-11</td>
<td>gfx-rs Ashes</td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metal</td>
<td>MoltenVK gfx-rs</td>
<td>clspv over MoltenVK?</td>
<td>MoltenGL Angle</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"'Vulkan everywhere'! Even if no native drivers on platform"

"Working towards ‘OpenCL Everywhere'!"

Vulkan is effective porting layer for API portability and stack simplification
Vulkan Portability on macOS and iOS

Vulkan Portability Extension provides query for available functionality.

Open source SDK on macOS
Build, run, and debug apps with validation layer support
https://vulkan.lunarg.com/

SPIRV-Cross
Convert SPIR-V shaders to Metal Shading Language

macOS / iOS Run-time
Maps Vulkan to Metal

Vulkan macOS SDK

All present functionality must be conformant to use logo

Vulkan applications shipping today on macOS or iOS using MoltenVK
ANARI - Analytic Rendering API

Scientific Visualization Portability
APIs to describe objects in a scene
- the renderer does the rest
Ray tracing was key motivation
- but can drive any renderer

Scene Graphs
SciViz Apps and Engines

Renderers:
Intel OSPRay, AMD ProRender, NVIDIA VisRTX etc.

Acceleration APIs:
Embree, Optix, Radeon Rays, CUDA, OpenCL, Vulkan etc.

Hardware:
CPUs, GPUs etc.
OpenXR - Cross-Platform Portable AR/VR

OpenXR is a collaborative design
Integrating many lessons from proprietary ‘first-generation’ XR API designs

* OpenXR 1.0 is focused on enabling cross-platform applications. Optional device plugin interface will be supported post V1.0

Working Group Participants

Games Engines
- EPIC
- unity

Desktop/Console VR
- HTC
- HP
- Huawei
- LG
- Microsoft
- oculus
- Valve

AR
- ARM
- Magic Leap
- SONY

Mobile VR
- Google
- Samsung
- Qualcomm
- Razer

UI Devices
- AMD
- Intel
- Qualcomm
- Valo
- ViewSonic

GPUs
- NVIDIA
OpenXR 1.0 Availability

Significant Community Feedback
- Improved Input subsystem, game engine editor support, loader ...

Provisional Specification
- GDC, March 2019

Ratify and Release
- OpenXR 1.0
- SIGGRAPH, July 2019

Finalize Conformance Test Suite
- Enable Officially Conformant Implementations

OpenXR for Windows Mixed Reality headsets and HoloLens 2
- PLUS extensions to support HoloLens 2 hand tracking, eye tracking, spatial mapping and spatial anchors

OpenXR support for Oculus Rift and Oculus Quest
- Oculus PC (for Rift) and Android SDK (for Quest)
- include OpenXR for native C/C++ development

‘Monado’ OpenXR open source implementation
- Support OpenHMD hardware and Nova Northstar AR HMD

OpenXR 1.0 plugin for Unreal Engine v4.2.4

Many more coming!
OpenXR is used with a 3D API

High-performance, low-latency 3D rendering and composition*
- Multiview
- Context priority
- Front buffer rendering
- Tiled rendering (beam racing)
- Variable rate rendering

Cross-platform access to XR
- HMDs and sensors
- XR application lifecycle
- Input device discovery and events
- Sensor tracking and pose calculation
- Frame timing and display composition
- Haptics Control

* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES
Wireless mobile device with display and sensors

OpenXR APIs hide the 5G round trip from applications

Display composition

Low latency Sensor Data

MEC (Multi-access Edge Computing) Server
1. Processes sensor data, including machine learning for environmental lighting, occlusion, scene semantics, object reconstruction and UI
2. Generates imagery from 3D models, including stereo, foveal rendering, ray-tracing, optics pre-distortion, varifocal processing

Generated Augmentations & Scenes

NVIDIA EGX

Needed assets loaded to edge server

Apps and 3D Assets

Location-aware Content Requests
**gltF Roadmap**

### glTF Universal Textures (imminent)
- Basis Universal encoding/transcoding
- KTX2 Container

- FlightHelmet_baseColor 2048 x 2048, RGB

### Second Generation Physically-Based Rendering (PBR)
- Set of coherent extensions
  - Clear coat (imminent)
  - Absorption/attenuation
  - Subsurface scattering
  - Anisotropy

- Inspiration from Dassault Systèmes Enterprise PBR Shading Model (DSPBR) and MDL
- Wide industry cooperation

### Seeking Requirements
- Subdivision surfaces
- Advanced Animation
- LOD and Streaming
- Compressed Point Clouds
- Cross-asset linking
- Enhanced Metadata
- Composability
- Instancing
- CAD/BIM model support
- Encryption and security
- 3D Printing

Working Group is constantly balancing feature requests against the ‘gltF Prime Directive’ - remain a universal and easy to process delivery format.
GPU COMPUTE APIS AT NVIDIA
Developer Choice

- Heterogeneous compute devices
- Cross-vendor open standard
- Simpler to program than rendering APIs

- CUDA-X libraries and tools
- Integrated HW/SW roadmap for rapid innovation
  - BUT GPU Only

- Widely available across multiple platforms
- Integrated rendering, data movement and compute
- Explicit hardware control
  - BUT GPU Only

Increasing layering, tooling and interop functionality enabling enhanced development and deployment flexibility
OpenCL is Widely Deployed and Used

**Desktop Creative Apps**
- Adobe
- Modo
- otoy
- CyberLink
- Autodesk
- GIMP
- ArcSoft
- Capture One
- Blackmagic Design
- blender
- SideFX
- Sony
- Altera
- AMD
- ARM
- Samsung
- NVIDIA
- Qualcomm
- STMicroelectronics
- Texas Instruments
- VeriSilicon
- Xilinx

**Parallel Languages**
- OpenACC
- SYCL
- OpenCL
- PyOpenCL
- GpuOpenCL

**Linear Algebra Libraries**
- SYCL-CLAS
- viennaCL
- CLBlast

**Machine Learning Libraries and Frameworks**
- OpenVINO
- MACE
- clDNN
- NNAPI
- SYCL-DNN
- caffe
- Caffe
- Acuity
- Halide
- VisionCpp
- OpenVX
- cuDNN
- tvm
- TIDL
- TensorFlow
- PyTorch
- Keras
- TensorFlow
- Caffe
- MXNet
- CHainer

**Vision and Imaging Libraries**
- MetaWare EV
- TI DL Library (TIDL)

**Machine Learning Compilers**
- plaidML
- tvm
- Wolfram Mathematica
- GNU Octave
- ArrayFire
- Matlab

**Math and Physics Libraries**
- Arm Compute Library
- Arm Compute Library
- NVIDIA GPU Compute Library
- AMD GPU Compute Library
- Intel GPU Compute Library
- ARM GPU Compute Library

**Hardware Implementations**
- AMD
- Intel
- NVIDIA
- ARM
- Samsung
- Qualcomm
- Texas Instruments
- VeriSilicon
- Xilinx

This work is licensed under a Creative Commons Attribution 4.0 International License
OpenCL Applications over Vulkan

- **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 for shipping production apps and engines on Android
  - Adobe Premiere Rush video editor - 200K lines of OpenCL C kernel code
  - Butterfly Network iQ Ultrasound on Android
  - Xiaomi MACE inferencing engine

[Diagram showing the flow of OpenCL C or C++ for OpenCL Kernel Sources through Clspv Compiler to OpenCL Application Host Code through Clvk run-time API Translator to Vulkan Runtime]

NVIDIA AND OPENCL
Active Investment and Support

Production-class OpenCL 1.2 on Linux and Windows
Active, ongoing improvements in performance and power efficiency for new architectures
Multi-GPU optimizations, multi-command-queue use case tuning
Optimized data transfers and GPU memory allocation

New Functionality
Half & Half2, arithmetic and conversions - in development

Expanding OpenCL Interop options
OpenGL and D3D9/10/11 - today. DX12 being planned
OpenCL/Vulkan interop - aiming for 2H20 - using standard Vulkan external memory interop hooks
NVIDIA driving Vulkan/OpenCL interop extension at Khronos

Providing guidance for deploying OpenCL apps over Vulkan
Leveraging the open source SPIR-V compiler
NVIDIA AND KHORRONOS API STANDARDS

NVIDIA has shipped Vulkan 1.2 and Ray Tracing drivers on day of specification release
NVIDIA chairing multiple Vulkan initiatives at Khronos: Ray Tracing, Machine-Learning, Vulkan Portability etc.
Increased Vulkan support in NSIGHT graphics development tools
https://developer.nvidia.com/Vulkan

Actively working to help evolve
OpenXR to complement VRWorks and enable EGX Edge Server AR
https://developer.nvidia.com/vrworks

Production-class OpenCL 1.2 on Linux and Windows
Active improvements in efficiency and performance
Vulkan Interop in development
https://developer.nvidia.com/opencl

NVIDIA initiated industry cooperation
Portable Analytic Rendering API
NVIDIA chairing the working group
https://www.khronos.org/anari
How To Get Involved!

• More information at other GTC recorded sessions
  - [S21770] NVIDIA Vulkan Features Update - including Vulkan 1.2 and Ray Tracing
  - [S22694] Ray Traced Reflections in Wolfenstein Youngblood

• Any company or organization is welcome to join Khronos!
  - For a voice and a vote in any of these standards - membership starts at $3,500

• OR request an invite to Vulkan, OpenCL, OpenXR Advisory Panels
  - No fee, execute Khronos NDA and IP Framework, provide requirements and spec feedback

• We welcome your feedback at NVIDIA or Khronos
  - Khronos Forums: https://forums.khronos.org/
  - Khronos Slack Channels: https://khronosdevs.slack.com/messages
  - Khronos open source GitHub repositories: https://github.khronos.org/

• Contact Neil Trevett
  - ntrevett@nvidia.com | @neilt3d | www.khronos.org