Khronos Overview

GDC | March 2016

Neil Trevett | Khronos President
NVIDIA Vice President Developer Ecosystem
ntrevett@nvidia.com | @neilt3d
Khronos Gaming API Update

• Vulkan 1.0 launched on 16th February 2016
  - The first new generation open standard graphics API since OpenGL - 25 years ago

• WebGL 2.0 is getting close
  - OpenGL ES 3.0 capability to the Web
  - Aiming for finalization in 2Q16

• glTF 1.0 momentum is building
  - Transmitting 3D assets to online Web, AR and VR experiences
The Need for a New Generation GPU API

- **Explicit**
  - Open up the high-level driver abstraction to give direct, low-level GPU control
- **Streamlined**
  - Faster performance, lower overhead, less latency
- **Portable**
  - Cloud, desktop, console, mobile and embedded
- **Extensible**
  - Platform for rapid innovation

OpenGL has evolved over 25 years and continues to meet industry needs - but there is a need for a complementary API approach.

GPUs are increasingly programmable and compute capable + platforms are becoming mobile, memory-unified and multi-core.

GPUs will accelerate graphics, compute, vision and deep learning across diverse platforms: FLEXIBILITY and PORTABILITY are key.
Next Generation GPU APIs

- DirectX 12: Only Windows 10
- Vulkan: Cross Platform
- SteamOS
- windows xp
- Windows 7
- Windows 8
- Windows 10
- Tizen
- Ubuntu
- Red Hat
- Android

© Copyright Khronos Group 2016 - Page 4
Vulkan - No Compromise Performance

- Potential Performance Gain

- Vulkan
  - Retains Traditional Binding Model
  - (but missing functionality such as Tessellation and Geometry Shaders)

- OpenGL
- OpenGL ES

Amount of work to port from traditional OpenGL and OpenGL ES
Vulkan Genesis

Khronos members from all segments of the graphics industry agree the need for new generation cross-platform GPU API

Significant proposals, IP contributions and engineering effort from many working group members

Including an unprecedented level of participation from game engine developers

18 months
A high-energy working group effort

Khronos’ first API ‘hard launch’ 16Feb16

Specification, Conformance Tests, SDKs - all open source...
Reference Materials, Compiler front-ends, Samples...
Multiple Conformant Drivers on multiple OS

Vulkan Working Group Participants
Vulkan Explicit GPU Control

Vulkan 1.0 provides access to OpenGL ES 3.1 / OpenGL 4.X-class GPU functionality but with increased performance and flexibility.

Vulkan Benefits

- Simpler drivers:
  - Improved efficiency/performance
  - Reduced CPU bottlenecks
  - Lower latency
  - Increased portability

- Resource management in app code:
  - Less hitches and surprises

- Command Buffers:
  - Command creation can be multi-threaded
  - Multiple CPU cores increase performance

- Graphics, compute and DMA queues:
  - Work dispatch flexibility

- SPIR-V Pre-compiled Shaders:
  - No front-end compiler in driver
  - Future shading language flexibility

- Loadable Layers:
  - No error handling overhead in production code
The Power of a Three Layer Ecosystem

Applications can use Vulkan directly for maximum flexibility and control

Application uses utility libraries to speed development

Utility libraries and layers

Application

Game Engines fully optimized over Vulkan

Applications using game engines will automatically benefit from Vulkan’s enhanced performance

Rich Area for Innovation
- Many utilities and layers will be in open source
- Layers to ease transition from OpenGL
- Domain specific flexibility

Similar ecosystem dynamic as WebGL
A widely pervasive, powerful, flexible foundation layer enables diverse middleware tools and libraries
Vulkan Multi-threading Efficiency

1. Multiple threads can construct Command Buffers in parallel. Application is responsible for thread management and synch.

2. Command Buffers placed in Command Queue by separate submission thread.

Can create graphics, compute and DMA command buffers with a general queue model that can be extended to more heterogeneous processing in the future.
Which Developers Should Use Vulkan?

• Vulkan puts more work and responsibility into the application
  - Not every developer will need or want to make that extra investment

• For many developers OpenGL and OpenGL ES will remain the most effective API
  - Khronos actively evolving OpenGL and OpenGL ES in parallel with Vulkan

Vulkan provides more choice to developers and can be used to create new classes of end-user experience
Vulkan Feature Sets

- Vulkan supports hardware with a wide range of hardware capabilities
  - Mobile OpenGL ES 3.1 up to desktop OpenGL 4.5 and beyond
- One unified API framework for desktop, mobile, console, and embedded
  - No "Vulkan ES" or "Vulkan Desktop"
- Vulkan precisely defines a set of "fine-grained features"
  - Features are specifically enabled at device creation time (similar to extensions)
- Platform owners define a Feature Set for their platform
  - Vulkan provides the mechanism but does not mandate policy
  - Khronos will define Feature Sets for platforms where owner is not engaged
- Khronos will define feature sets for Windows and Linux
  - After initial developer feedback
Vulkan Window System Integration (WSI)

- Explicit control for acquisition and presentation of images
  - Designed to fit the Vulkan API and today’s compositing window systems
  - Cleanly separates device creation from window system

- Platform provides an array of persistent presentable images = Vulkan Swapchain
  - Device exposes which queues support presentation
  - Application explicitly controls which image to render and present

- Standardized extensions - unified API for multiple window systems
  - Works across Android, Mir, Windows (Vista and up), Wayland and X (with DRI3)
  - Platforms can extend functionality, define custom WSI stack, or have no display at all
SPIR-V Transforms the Language Ecosystem

- First multi-API, intermediate language for parallel compute and graphics
  - Native representation for Vulkan shader and OpenCL kernel source languages
- GL_KHR_vulkan_glsI spec released - adds the GLSL features needed for Vulkan
  - Descriptor sets, push constants, specialization constants
  - Separate images/samplers, sub pass input images...
  - Updated front-end open source compiler in Khronos GitHub

Multiple Developer Advantages

Same front-end compiler for multiple platforms
Reduces runtime kernel compilation time
Don’t have to ship shader/kernel source code
Drivers are simpler and more reliable
Evolution of SPIR Family

- SPIR-V is first fully specified Khronos-defined SPIR standard
  - Does not use LLVM to isolate from LLVM roadmap changes
  - Includes full flow control, graphics and parallel constructs beyond LLVM
  - Khronos will open source SPIR-V <-> LLVM conversion tools

<table>
<thead>
<tr>
<th></th>
<th>SPIR 1.2</th>
<th>SPIR 2.0</th>
<th>SPIR-V 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLVM Interaction</td>
<td>Uses LLVM 3.2</td>
<td>Uses LLVM 3.4</td>
<td>100% Khronos defined Round-trip lossless conversion</td>
</tr>
<tr>
<td>Compute Constructs</td>
<td>Metadata/Intrinsics</td>
<td>Metadata/Intrinsics</td>
<td>Native</td>
</tr>
<tr>
<td>Graphics Constructs</td>
<td>No</td>
<td>No</td>
<td>Native</td>
</tr>
<tr>
<td>Supported Language Feature Set</td>
<td>OpenCL C 1.2</td>
<td>OpenCL C 1.2 / 2.0</td>
<td>OpenCL C 1.2 / 2.0 GLSL</td>
</tr>
<tr>
<td>OpenCL Ingestion</td>
<td>OpenCL 1.2 Extension</td>
<td>OpenCL 2.0 Extension</td>
<td>OpenCL 2.1 Core</td>
</tr>
<tr>
<td>Vulkan Ingestion</td>
<td>-</td>
<td>-</td>
<td>Vulkan Core</td>
</tr>
</tbody>
</table>
Driving the SPIR-V Open Source Ecosystem

SPIR-V Tools
- SPIR-V Validator
- SPIR-V (Dis)Assembler
- LLVM to SPIR-V Bi-directional Translator
- SPIR-V (Dis)Assembler
- SPIR-V Validator
- OpenCL C
- OpenCL C++
- GLSL
- Third party kernel and shader Languages
- Other Intermediate Forms

SPIR-V
- 32-bit Word Stream
- Extensible and easily parsed
- Retains data object and control flow information for effective code generation and translation

Khronos will open source these tools and translators

OpenCL Runtime

IHV Driver Runtimes

OpenCL

GLSL
Vulkan Developer Resources at Launch

---

Khronos.org
Canonical Resources
Specifications, Header Files
Feature Set Definitions
(Windows and Linux - post developer feedback)
Quick Reference and Reference Pages
Conformance Test Source and Test Process

Materials to Build SDKs and Tools
Compiler toolchain sources
Validation Layer Source
Loader Source
Layers and Loader documentation
(open source resources in github.com/KhronosGroup)

---

LunarG
Windows and Linux Installable SDKs
Loader and Validation Layer binaries
Tools Layers - source and binaries
Samples - source and binaries
Windows get started guide

---

IHV Websites
Drivers and Loader
Vendor tools and layers

Third Party Websites
Layers, Samples etc.
Vulkan Tools Architecture

- Layered design for cross-vendor tools innovation and flexibility
  - IHVs plug into a common, extensible architecture for code validation, debugging and profiling during development without impacting production performance

- Khronos Open Source Loader enables use of tools layers during debug
  - Finds and loads drivers, dispatches API calls to correct driver and layers
LunarG SDK for Vulkan

- Valve sponsored LunarG to develop a free, open source SDK for Vulkan
  - Utilities, samples, debugging tools, documentation
  - For Windows and Linux on launch - Android coming soon

- Validation Layer - checks many aspects of Vulkan code:
  - Device limits, draw state, parameter values
  - Multi-thread object access rules, texture and render target formats
  - Object Tracker, Memory Tracker

- Other SDK Tools
  - Trace and replay tools
  - GLSL Validator
  - SPIR-V Disassembler and Assembler

- RenderDoc Graphics debugger
  - Free and open source
  - Adding Vulkan support
  - https://github.com/baldurk/renderdoc
Conformant Vulkan Drivers

- 30 Driver submissions passed conformance at Vulkan 1.0 launch
  - www.khronos.org/conformance/adopters/conformant-products
  - AMD: Linux and Windows
  - ARM: Linux
  - Imagination Technologies: Linux
  - Intel: Linux, Windows
  - NVIDIA: Android 6.0, Linux (desktop and embedded), Windows 7-10
  - Qualcomm: Android 6.0

Khronos and Android leveraging and merging Vulkan tests with the Android Open Source Project (AOSP) and DrawElements Quality Program (dEQP) framework. Open source Vulkan conformance test suite hosted on GitHub. Implementers must pass Test Suite. Khronos administered Vulkan Adopters Program. Implementations that pass test process may use Vulkan trademark.
“Vulkan has a huge potential! We’re only scratching the surface of what can be done with it, and porting The Talos Principle to Vulkan should be seen as a proof of concept,” said Dean Sekulic graphics engine specialist at Croteam. “Vulkan in just one sentence? The endless war between performance and portability is finally over!"

“Talos Principle on Steam has beta Vulkan back-end

“By building your application or game using the Vulkan API, you can run your modern graphics application or game unchanged across an entire industry of platforms and development tools”

Brenwill Workshop

Vulkan and OpenGL ES over Metal - in development
One Week After Launch of Vulkan 1.0

Valve’s SteamOS now supports Vulkan, the cross-platform alternative to DirectX 12

https://www.youtube.com/watch?v=FnKu7MLB7vQ

http://blogs.vk.com/blog/2016/02/22/shield-marshmallow-vulkan/

Of the many things Samsung announced with the Galaxy S7 and Galaxy S7 edge, the inclusion of Vulkan APIs has the potential to be the most significant.

Ubuntu 16.04 LTS to Ship with Full Support for Vulkan in Mir Display Server

Canonical is jumping on the Vulkan train
Feb 17, 2016 18:49 GMT - By Shane Santos

Ubuntu 16.04 LTS (Xenial Xerus) is going to integrate full support in Mir for the latest Vulkan 1.0 specifications.

Vulkan is stealing all the headlines in the Linux world and with good reason. It’s an incredible leap forward for the open source platform, even if Vulkan is technically aimed at all the major operating systems, including Windows, Android, and even Tizen.

http://www.androidauthority.com/powervr-series8xe-673991/

http://www.androidcentral.com/vulkan-samsung-galaxy-s7-potentially-very-big-deal

http://www.androidcentral.com/vulkan-samsung-galaxy-s7-potentially-very-big-deal

http://www.androidauthority.com/powervr-series8xe-673991/


http://www.androidcentral.com/vulkan-samsung-galaxy-s7-potentially-very-big-deal


http://www.androidauthority.com/powervr-series8xe-673991/


http://www.androidauthority.com/powervr-series8xe-673991/

http://blogs.vk.com/blog/2016/02/22/shield-marshmallow-vulkan/
WebGL Games Ecosystem

>1 Billion desktop and mobile browsers
Including Facebook social gaming
**WebGL 2.0**

Enhanced visual quality, performance
- Instancing  |  Multiple render targets  |  Uniform buffers
- Transform feedback  |  Multi-sampled Renderbuffers  |  3D textures
- NPOT textures  |  More texture formats  |  Occlusion queries
- Vertex array objects  |  Sampler objects  |  Sync objects
- Fragment depth  |  Primitive restart  |  ...

**OpenGL ES 2.0**

- 32-bit integers and floats
- NPOT, 3D/depth textures
- Texture arrays
- Multiple Render Targets
- Compute Shaders

**OpenGL ES 3.x**

- Tessellation and geometry shaders
- ASTC Texture Compression
- Floating point render targets
- Debug and robustness for security
- Epic’s Rivalry demo using full Unreal Engine 4
  https://www.youtube.com/watch?v=jRrG95GdaM
3D Needs a Transmission Format!

- Efficient Format to define, transmit and use 3D Assets
  - Rich scene/object representation with animations, materials - OBJ, STL etc. too limited
  - Common format for tools, engines to import and export
  - Publishing format for content repositories and services

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>H.264</td>
<td>JPEG</td>
<td>?</td>
</tr>
<tr>
<td>napster</td>
<td>YouTube</td>
<td>facebook</td>
<td>!</td>
</tr>
</tbody>
</table>

A widely adopted media format ignites previously untapped commercials opportunities.
glTF = “JPEG for 3D”

- ‘GL Transmission Format’
  - Runtime asset format for WebGL, OpenGL ES, and OpenGL applications

- Compact representation for download efficiency
  - Binary mesh and animation data

- Loads quickly into memory
  - GL native data types require no additional parsing

- Full-featured
  - 3D constructs (node hierarchy, materials, animation, cameras, lights)

- Runtime Neutral
  - Can be created and used by any tool, app, or runtime

- Flexible Extensibility
  - E.g. payloads with compression and streaming
glTF Internals

- **JSON describes node hierarchy**
  - Includes cameras
  - References geometry, animations, skins, shaders, textures

- **Vertices**
  - Uses native typed array format
  - Includes key-frame animations and skinning

- **Shaders**
  - With extensions for materials

- **Textures**
  - Use existing standard image compression formats e.g. JPEG

- **Extras**
  - For app-specific data (metadata)
glTF Ecosystem

Tools

Translators

Validator

Engines

- Autodesk AssImp
  Autodesk FBX -> glTF
- COLLADA2GLTF
- Cesium converter
- OBJ2GLTF
- glTF Pipeline

Blender DIRECT export

Convert | Optimize

Validate

Request for Quotations (RFQ) to create glTF Validator issued today!

glTF Ecosystem Page
https://github.com/KhronosGroup/gltf#gltf-tools
glTF for VR and AR

Direct Perception

Location and Sensors

Augmentation Assets

Easing the authoring of 3D assets

Optimizing 3D Asset Delivery over a network

Avoiding ‘Silo’d’ content in proprietary formats

Authored Assets and Experiences
glTF Validator RFQ! Consider Bidding!!

- Validate glTF assets: geometry, material, animation, and skins
  - Ensure compliance with spec and schema
- Augment glTF sample models to cover corner cases
- Optional features
  - Binary glTF, REST service, drag-and-drop validator, drag-and-drop three.js viewer

- Schedule
  - March 15 - Khronos Released RFQ
  - March 31 - RFQ responses received by Khronos
  - April 15 - Contractor selected and notified
  - April 29 - Contract executed and start of work

- [https://www.khronos.org/rfq](https://www.khronos.org/rfq)
Khronos Roadmap Discussions

SPIR-V Ingestion for OpenGL and OpenGL ES for shading language flexibility

Khronos members decide how to evolve and mix and match a rich set of APIs and technologies to meet market needs

OpenCL-class Heterogeneous Compute to Vulkan runtime:
- C++ Shading Language
- Tiered precision
- Shared virtual memory
- Dynamic parallelism...

Bring new generation API capabilities to WebGL once they become pervasive
Khronos Open Standards for Graphics and Compute

1990’s - Workhorse cross-platform professional 3D apps & gaming

2000’s - Ubiquitous mobile gaming & graphics apps

2005 - Safety Critical Graphics

2008 - Heterogeneous parallel compute

Portable intermediate representation for graphics and parallel compute - 2014

High-efficiency GPU graphics and compute for performance critical apps - 2016

LATEST STATUS

- New Extensions to enable latest desktop graphics capabilities
- OpenGL ES 3.2 released to bring AEP functionality to core
- New Safety Critical Working Group - Call for Participation
- OpenCL 2.0 specification update and C++ Headers released
- Provisional Spec Update and significant open source activity
- Adopted by Android and other platforms. Building ecosystem