

GDC[®] 17

THE
T·A·L·O·S
PRINCIPLE



Vulkan Game Development on Mobile

KHRONOS[™]
GROUP
CONNECTING SOFTWARE TO SILICON

Alen Ladavac, CTO, Croteam



@AlenL



alenl@croteam.com

Vulkan[®]



*Croteam is the oldest and best known Croatian game development studio responsible for the famous franchises **Serious Sam** and **The Talos Principle**.*



Vulkan's Promises

- Thinner driver
 - Less overhead
 - Better drivers across vendors
- Control over PSOs and command buffers
 - You control when caching happens!
- Direct control over memory use
 - Again – bounded performance
- Native multithreading

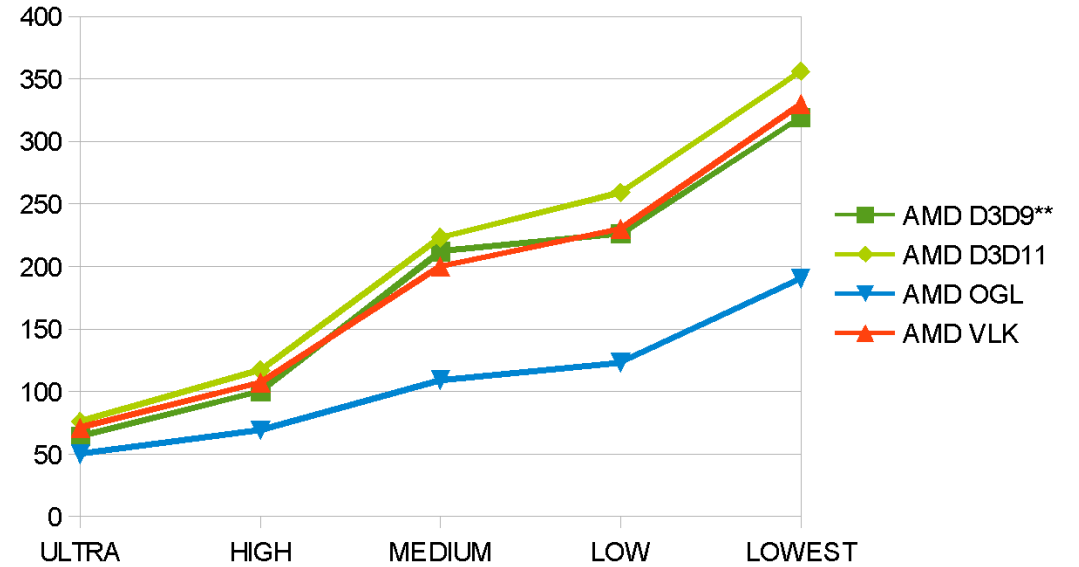
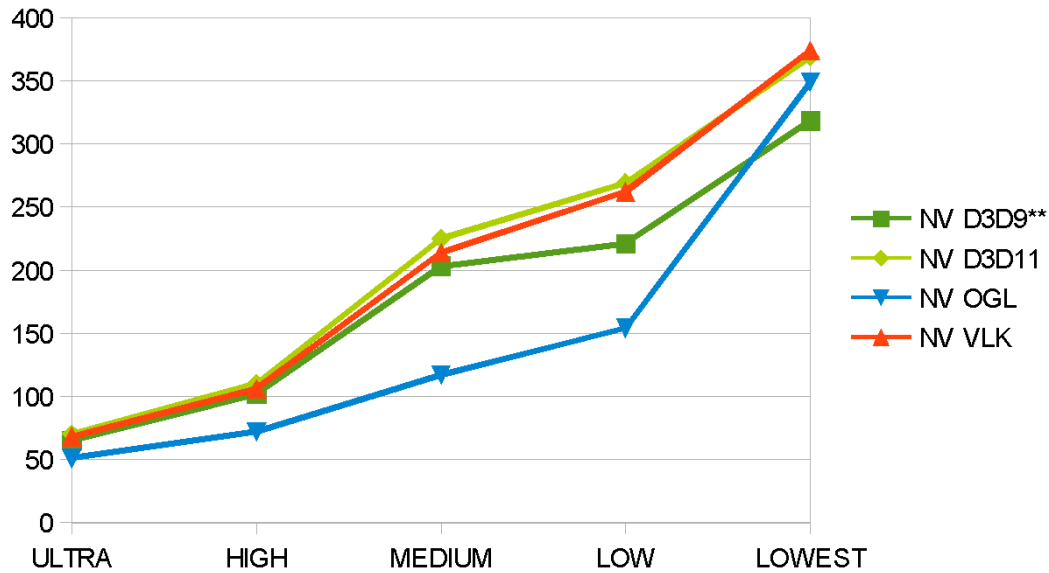
Traps, Pitfalls and Lifesavers

- Traps&pitfalls:
 - Synchronization (barriers, fences...)
 - Memory (alignments, types, block limits...)
 - Rendering passes, buffer layouts...
 - Different behavior across devices (all of above)
- Your friends:
 - Validation layers (Use them!)
 - RenderDoc, NSight, GPU PerfStudio

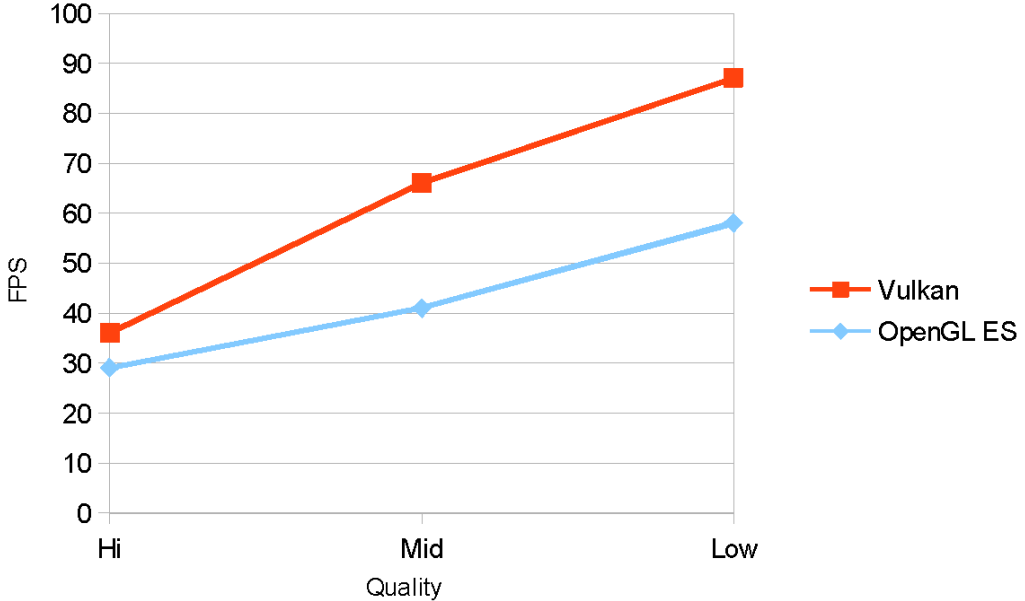
“Quick Port” approach

- GLSL stays mostly the same
- Implement PSO caching (on demand)
- Emulate “binding” model for PSOs and DSs
- Explicit memory allocator
- Explicit uploading of resources
- Barriers for synchronization (e.g. Read-after-Write)
- 4 man-months (including moving target!)

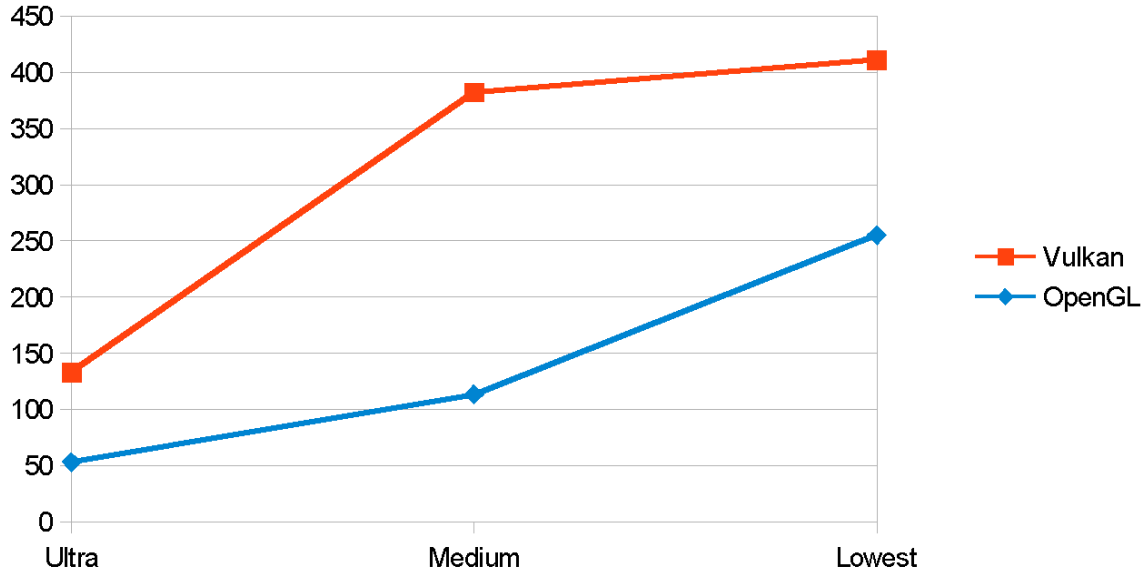
Did it deliver? (Windows)



Did it deliver? (Android+Linux)



Android
(NVIDIA Shield TV)



Android
(NVIDIA Shield TV)



Performance observations

- Current performance is already promising
 - Better than DX9 and OpenGL, mixed results vs DX11
- Already the API of choice for Linux and Android
- Room for improvements (huge!):
 - Pre-cache PSOs
 - Prepare materials in advance
 - Native multithreading
 - Expected to be much faster than DX11 on Windows

Still working on...

- Native multithreading (hard!)
 - Pre-recorded buffers are all or nothing!
 - Have to know layout states during recording!
- Precompiled PSOs
 - Some states are very impractical being static
 - eg CW/CCW winding flag \Rightarrow all states x2 due to mirrors
 - but that's how (some) hardware needs it

The Drivers

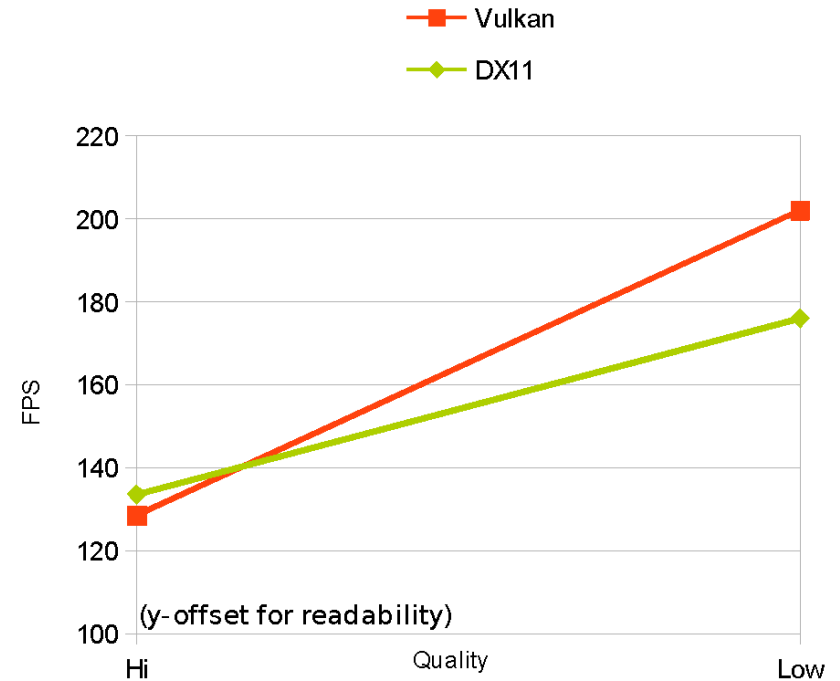
- Performance ?
 - Check!
- Stability?
 - Aaaalmost there... :)
 - Still more problems than in DX11 and GLES:
 - Crashes and rendering artefacts
- Android infrastructure issues
 - Hard and slow to get driver fixes out.

Validation Layers

- Very useful, but beware:
 - False positives still happen
 - False negatives still happen
 - They both hurt your development time.
- Still more good than harm!
- This improves over time as checks are fixed and added.

What's with fill rate?

- Suspects:
 - GLSL itself?
 - Our GLSL implementation?
 - Waiting for HLSL to SPIRV
 - Some cornercases that we use are not supported
 - When we can use it, we will know what's the cause of this paradox.



Is it really that grim?

- No.
 - It's come a long way, and everybody is on board.
 - Set to be the cross-platform solution for the future.
 - On Android, it looks close to being more useful than GLES
 - Already faster by a large margin!
 - “Just” needs driver fixes.
 - (But this mostly concerns more complex rendering systems!)

When to use it?

- If you need top performance
and
- If you are CPU bound
and
- If you can afford a more complex implementation
(or if the game is not very diverse in rendering requirements)

GDC[®] 17

THE T·A·L·O·S PRINCIPLE



Thanks!

KHRONOS[™]
GROUP
CONNECTING SOFTWARE TO SILICON

Alen Ladavac, CTO, Croteam



@AlenL



alenl@croteam.com

Vulkan[®]



*Croteam is the oldest and best known Croatian game development studio responsible for the famous franchises **Serious Sam** and **The Talos Principle**.*

