A Vulkan Video Encoder from Mesa to GStreamer

Hyunjun Ko / Stéphane Cerveau

2024-01-06
Implementing a Vulkan Video encoder from Mesa to GStreamer

Stéphane Cerveau, Igalia
Hyunjun Ko, Igalia
Agenda

1. Vulkan Video
2. Mesa - driver
3. GStreamer - application
4. Demos
Vulkan Video

- Stateless codecs using GPU hardware acceleration
- Supported codecs: H.264, H.265, AV1
- Closer integration with Graphics and Displays.
- Cross-platform and vendor-neutral low-level HW stateless video codecs API
  - Each driver can operate differently depending on its capabilities with a common API.
Vulkan Video Timeline

- **March 2018**: TSG was created and driven by IHVs such as AMD/Intel/Nvidia and open source operators
- **April 2021**: **Provisional extensions released** including the Video Decode and Encode extensions
- **January 2023**: Video Extensions for Accelerated H.264 and H.265 Decode **released**
- **December 2023**: Khronos **finalized** Vulkan Video Extensions for Accelerated H.264 and H.265 Encode
Encoding basics
The status of Vulkan Video Encoder Support in Mesa project
Contents

- What is Mesa?
- History of Vulkan Video development in Mesa
- Drivers supporting Vulkan Video in Mesa
- Development of Vulkan Video Encoding on Intel GPUs
- Challenges
- Plan
What is Mesa?

- **https://mesa3d.org/**
- Began as an open source implementation of the OpenGL.
- Now actively implementing Vulkan specification on various GPUs.
  - Intel, AMD, Qualcomm Adreno(R/E), Raspberry PI, etc..
  - **https://gitlab.freedesktop.org/mesa/mesa**
- Contributors: Igalia, Intel, Google, Collabora, Mesa community.
History of Vulkan Video development in Mesa

- Dave Airlie started in 2022 on AMD and Intel GPU (RADV and ANV)
  - Dave's blog post
  - With Lynne on FFmpeg.
  - Igalia joined on GStreamer later.
- Hyunjun joined in 2023, started working on Intel GPU (ANV)
- So Dave fully dedicated to AMD GPU (RADV).
The status of Vulkan Video development in Mesa

- Implemented and landed decoder first for h264 and h265.
  - ANV H264 MR #20782
  - ANV H265 MR #22202
  - RADV H264/265 MR #20388
- Now MR for h264/5 encoding are almost ready.
  - RADV MR
  - ANV Branch
Drivers supporting Vulkan Video in Mesa

<table>
<thead>
<tr>
<th>GPU</th>
<th>H264 dec</th>
<th>H265 dec</th>
<th>H264 enc</th>
<th>H265 enc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel(ANV)</td>
<td>O</td>
<td>O</td>
<td>WIP</td>
<td>WIP</td>
</tr>
<tr>
<td>AMD (RADV)</td>
<td>O</td>
<td>O</td>
<td>Ready</td>
<td>Ready</td>
</tr>
</tbody>
</table>
Working on Intel GPUs (1)

- Dived into Intel Vulkan driver (ANV) in 2023.
  - Started working on H265 decoding first.
- Tons of documents and source code of Intel VAAPi drivers.
  - Exhausted to learn lots of Video commands.
  - But better than nothing :)

14
Working on Intel GPUs (2)

- H265 Encoding Sequence
Working on Intel GPUs (3)

- Complete each command very carefully.
  - Otherwise you got a GPU hang or even whole system down.
- When you get a GPU hang and don't see any clue.
  - Dumping whole video commands encoding a frame (by VAAPI Driver) into a file.
  - Compare to commands that you created
Working on Intel GPUs (4)

- Thanks to the existing infrastructures of ANV
  - Easy to handle memories and images.
- Thanks to ANV maintainers.
  - They actively reviewed relevant merge requests.
Co-working with Applications

- GStreamer, VK CTS, FFMpeg...
- Each uses different parameters and makes it find bugs easily.
  - Different resolution, profile, SPS, PPS parameters.
Challenges

- GPU hang.
  - Not enough useful tools to investigate.
- Lots of generations of Intel GPUs.
  - Different commands, parameters, memory size, alignment, etc...
Plan 2024

- Land h264/h265 encoding support.
- AV1 support.
- Support other GPUs?
• a 20-year-old framework for streaming media applications.
• Black boxes interconnection system
• Native, multiplatform, highly-optimized framework
GStreamer pipeline
Vulkan Video support

- Follow **Vulkan Video Status**
- Vulkan H.264 decoder merged in December 2023
- Vulkan H.264/H.265 encoder under review.
State machine

- Init video session (H.264, H.265)
- Init session params (SPS etc.)
- Reset codec State
- (Change quality rate control)
- Retrieve session params
- Set slice header
- Encode buffer Begin/Encode/End
- Query encode result
Challenges
Cross platform API

- Hardware crashes, Thanks Validation Layers!
- Exact behavior varies by hardware vendor
- Rate control and quality issues
Synchronization

• Major issues with both decoder and encoder
  ○ Old memories from Vulkanised 2023, green screen...
  ○ Rework of GStreamer state machine with memory barriers, fences.
  ○ GstVulkanOperation to handle commands synchronization.
DPB management

- Understand the correct use of Begin and Encode reference slots.
  - Need to declare the reference within Begin command and use it during the Encode command.
- Various crashes in drivers not detected by the Validation Layers when the standard H.26x parameters (SPS, slice header) were not filled properly.
- Vulkan reference slots management.
Vulkan tooling

- Validation Layers:
  - Help validate we understand the specifications correctly
  - Do not prevent a misconfiguration of the std parameters
  - Mesa drivers helped to understand driver's pitfalls when VL was clear.
- GFXReconstruct, VK_LAYER_LUNARG_api_dump layer.
- CTS: Help with a reference design reviewed by IHVs.
Demos
Questions ?
Thanks

Join us!

https://www.igalia.com/jobs