Vulkan for Wayland Compositors: the new best choice?

Austin Shafer

Why Vulkan?

Vulkan's explicit design can be advantageous for implementing compositors

Who this talk is for:

- New compositors and compositors with multiple supported APIs

- Not suggesting:

  - Every compositors should switch

  - Vulkan is a perfect silver bullet

- Goal is to promote discussion!

NVIDIA

Explicit Image Properties
Major Advantage

- Image residency and tiling are core parts of Vulkan
    - vkCmdCopyImage() and vkCmdBlitImage can perform format conversions
    - No need to invoke rendering

NVIDIA

Explicit Image Properties
Major Advantage

- Image residency and tiling are core parts of Vulkan
  - vkCmdCopyImage() and vkCmdBlitImage can perform format conversions
  - No need to invoke rendering
- Cross-vendor copies heavily involve converting formats
  - Allocate and copy to a linear format, import linear buffer on second GPU

Explicit Image Properties
Major Advantage

- Image residency and tiling are core parts of Vulkan
  - vkCmdCopyImage() and vkCmdBlitImage can perform format conversions
  - No need to invoke rendering
- Cross-vendor copies heavily involve converting formats
  - Allocate and copy to a linear format, import linear buffer on second GPU
- Advantageous for vendors which cannot render to linear
  - Currently with EGL we have to fall back to a very expensive CPU copy
  - With Vulkan we can still use vkCmdCopyImage()

Multi-Device Awareness

Major Advantage

Devices in Vulkan

- Physical device enumeration, Device Groups, VkQueues, etc

- No need to for tracking current context

- Memory Residency is Explicit

Potential for Spec Extension

- Multi-Vendor Device Groups?

- VK_IMAGE_USAGE_SCANOUT_BIT_EXT?

  - Support might have to be advertised through the swapchain

  - EGL sort of has this with EGL_MESA_drm_image

- Cross GPU copies in an API call?

  - Maybe this doesn't belong in a graphics API

NVIDIA

Miscellaneous
Minor Advantages

- Well-defined synchronization and memory model

  ○ No need to rely on Implicit Sync or glFinish()

- Fine-grained work submission

- Validation Layers

- Timeline Semaphores can be powerful

  ○ Could pair well with linux-drm-syncobj-v1 if we provide a way to import a DRM syncobj?

NVIDIA