

Workshop recap: Vulkan “Gallium”

Faith Ekstrand

DYNAMIC STATE IS HARD



I KNOW



Workshop recap: Vulkan “Gallium”

- People hate Gallium caps
 - It's useful for servicing multiple APIs
 - It's hell to try and figure out why your extension isn't getting enabled
 - People would like to expose Vulkan features more directly
 - But maybe not extensions?

Workshop recap: Vulkan “Gallium”

- People hate Gallium caps
- Concerns about vfunc overhead
 - More layers = more vfuncs
 - GL → Gallium → Zink → Vulkan loader → Shiva → Driver is a lot
 - Maybe we can mitigate this somehow?
 -

Workshop recap: Vulkan “Gallium”

- People hate Gallium caps
- Concerns about vfunc overhead
- Shader objects?
 - Some people are concerned about going full SSO
 - Shiva will try to degrade gracefully based on what the client uses
 - It will look like shaders
 - If the client uses pipelines, the driver still gets everything



Workshop recap: Vulkan “Gallium”

- People hate Gallium caps
- Concerns about vfunc overhead
- Shader objects?
- Descriptors?
 - We need to support all 3 models, probably
 -

Workshop recap: Vulkan “Gallium”

- People hate Gallium caps
- Concerns about vfunc overhead
- Shader objects?
- Descriptors?
- Dynamic state? CSOs?
 - This needs its own slide

Dynamic state and CSOs

- Some sort of state grouping is good for save/restore
- Drivers generally don't want hyper-granular state updates
- Everyone groups state differently
 - Gallium CSOs are good but they're not perfect
- Turnip actually likes baking state in pipelines
- State is hard?