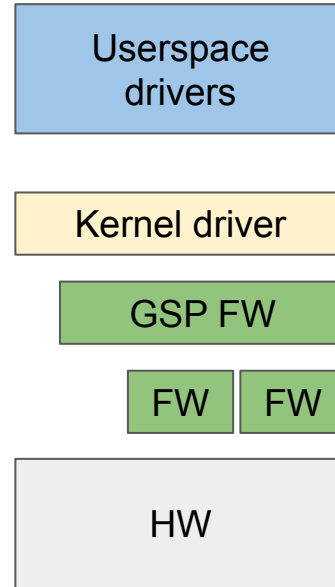
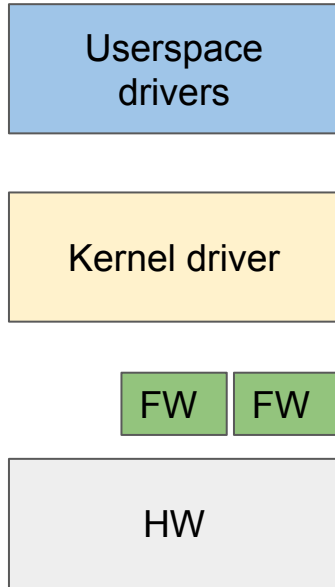


Nouveau GSP GPU VA management

Dave Airlie
Red Hat Distinguished Engineer

airlied@redhat.com

What is GSP?



GSP Pros/Cons?

- Pros
 - Reclocking is possible
 - Same firmware as NVIDIA uses
- Cons
 - No stable ABI
 - 100s of RPCs not really documented
 - Large firmware files
 - (/boot and initramfs sizes)

Nouveau + GSP current status

- Refactoring and preparation
- Initial GSP support for one firmware
 - Merged for 6.7-rc1
- Missing features
 - Fault handling
 - Sensor monitoring
- Future features
 - Dynamic ABI generation (rust?)

GPU memory management - history

- VRAM/GTT
- Kernel relocations
- Virtual memory
 - per-context/process

GPU memory management

- GEM for buffer object management
- TTM for discrete VRAM buffer object management
- syncobjs/fences for synchronising buffer operations
- Initial VA in-kernel tied to buffer object
 - Sufficient for OpenGL
 - Not future proof

Vulkan requirements

- Vulkan introduces sparse memory
 - Userspace VA management
 - Sync and async (pipelined) VA updates
- Drivers started inventing VA management
- VM_BIND

Common code for acceleration

- Modesetting framework/atomic
- Accel common code
 - Scheduler
 - TTM
- GPU VA management

GPU Virtual Memory Manager - GPUVM

- Inspired by amdgpu code
- Hopefully useful for all drivers
 - Nouveau, xe, panfrost
- Porting possibilities
 - amdgpu, msm
- Many iterations
 - Tried using maple trees

The great fence signaling critical section

- dma-fence waits have to be bounded
 - Memory management deadlocks otherwise
- Can be called from the shrinker
- Limits operations in certain fence signalling critical sections
 - Like memory allocations
 - Always using GFP_ATOMIC not a great plan

Nouveau: current status

- Initial VM_BIND UAPI
 - GPUVM
 - Syncobj + scheduler integration
 - Upstream in 6.6
- Improvements to gpubm/scheduler
 - In progress for 6.8

Userspace

- NVK project - Vulkan driver for nouveau
- Initial bringup using old codegen compiler
- NAK - New compiler backend
 - Merged into mesa master last night
 - Running much faster than codegen
- Close to Vulkan 1.0/1.1 conformance on Turing

Questions/Demo?