Virgil3D: a virtio based 3D GPU

Dave Airlie
<airlied@redhat.com>
Red Hat, Brisbane.
Virgil3D

- Research project
  - Initial implementation just to see
- Focus
  - Work out the 3D side of a virtio GPU
  - Get familiar with virtio and qemu code
- Based on Mesa project Gallium 3D
- Linux focused
Other projects

- **Vgallium**
  - relied on gallium drivers in host
  - Old version of gallium
  - Unmaintained

- **VirtualBox GL passthrough**
  - GL based is too large a surface area
  - Unknown security implications

- **Vmware SVGAII**
  - Closed source
  - Based on DX9 so limited capability
Virtio interface

- Single virtio ring
  - Context management
    - Create, Destroy, Attach Resource, Detach Resource
  - 3D Resource management
    - Create, Destroy, Flush, Attach SG, Detach SG
  - DMA-like transfer operations
    - Get, Put
  - Modesetting
  - Command stream submission
  - Capabilities
  - Fencing
- IRQ for fencing
- Config space for fencing and cursor handling
Rendering Command stream

• Gallium state objects
  – Blend, rasterizer, dsa, shaders, samplers, queries etc.
  – Create, bind, destroy operations

• Non-state operations
  – Framebuffer, scissor, viewport, vbos.

• Rendering
  – Draw, clear, blit

• Queries
Renderer

- Convert gallium states to GL interface
- Convert TGSI shaders to GLSL shaders
- GL host context per guest context
  - Required for proper conditional rendering operation
- Works out capabilities from GL version and extensions
- Currently uses GLSL 1.30 shader programs
GL versions

- Guest currently at GL 2.1 + GLSL 1.20
- Host requires GL2.1 + GLSL1.30
- Up to GL3.0 in the guest mostly done
  - Issues with multisample textures and hibernate/migration
- 3.1 and above open a number of questions
  - Lack of ARB_compatibility
Issues

• How best to get 64-bit values back from host
  - Status page seems like my best answer
• How to get fence irqs?
  - Second vq instead – seems like overkill
  - Can a vq attach 0 elements?
  - Or maybe just attach status page all the time
• GL 3.x context creation
  - Due to deprecated features
virtio-gpu

- Secondary project
- Produce a basic virtio gpu that the virgil renderer can attach to.
- Multi-head capable
- Unaccelerated
- PCI and VGA extras
Port QEMU to SDL 2

- SDL 2
  - Multiple window support
  - ARGB cursor support
  - Better GL support
    - EGL
  - Very different input
Qemu console multi-head

• Initial implementation
  – Add arrays of DisplaySurfaces to QemuConsole
  – Add _idx version of some interfaces
  – Use SDL2 multi-window support for demo

• TODO
  – Howto to work out num heads limits
Beyond SDL

- Libvirt integration
- Security
- DRM render nodes
- Using EGL and dma-buf to share the final rendered image
- Viewer using GLX/EGL to composite final rendered image
Demos

- Virgil3D rendering

- Virtio-vga multi-head