An Introduction to Intel GVT-g (with new architecture)

Zhi Wang  
zhi.a.wang@intel.com  
WW24’17
GPU Virtualization

VM

VM

...

VM

Hypervisor

Intel Processor Graphics
Existing Arts vs Intel GVT-g

Legacy VGA Emulation

- 2D only

- Qemu
- VGA

Hypervisor

GPU

API Forwarding

- BE
- GFX
- FE
- APIs

Hypervisor

GPU

Direct Pass-Through

- GFX

Hypervisor

GPU

Full GPU Virtualization

- vGPU
- GFX

Hypervisor

GPU

- +3D
- +Sharing
- - Media/compute
- - Compatibility

- +3D/media/compute
- +Performance
- +Sharing

- +3D/media/compute
- +Performance
- +Sharing
Near native performance with direct GPU execution

Run native graphics stack to sustain visual experience

Accelerate up to 8 VMs plus Dom0
GVT\textsubscript{g} (KVMGT) Architecture

- **Host Linux**
  - Libvirt
  - Mdev mgmt
  - Qemu
    - VFIO

- **VFIO**
  - Mdev Framework

- **GFX Driver**
  - vGPU
  - Device Model

- **IOMMU Driver**

- **Guest OS**
  - GFX Driver

- **KVM**

- **IOMMU**

- **GPU**
Resource Management

- vGPU resources
  - Graphics memory
  - Fence registers

- Request GPU resource from host resource allocator

- GVT manages vGPU resource according to vGPU types
Interrupt

Full GPU interrupt virtualization

• Display Interrupts
  – VBlank
  – GMBUS
  – AUX Channel

• GPU command Interrupts
  – MI_USER_INTERRUPT
  – PIPE_CONTROL_NOTIFY

• Context Switch Interrupts

GPU Interrupts

GVT Interrupt Emulation Policy Framework

Interrupt Emulating

Virtual Interrupt Regs

MPT

Host

VM
Shadow GPU Page Table - GGTT

- Global graphics memory space (GGTT) is partitioned
  - Dedicated resource for each VM

[Diagram showing Host View, VM1, VM2, VM3 with High gfx memory space, Low gfx memory space, Ballooned graphics memory space, Available graphics memory space to VMs]
Shadow GPU Page Table - PPGTT

Features

• 2/3/4 level page table
• True per-process PPGTT
• Page table cache
• Reference counting
• Out-of-Sync shadow
Scheduling

- vGPU 1
- vGPU 2
- vGPU 3...

Virtual HW Submission Interface

Scheduling Policy Framework

Workload Scheduler

Host i915 Submission Interface

- vExeclist
- vGuC
- vGPU workload Q
- GVT-g Scheduling
- i915 scheduling
Scheduling

I915 request
I915 request

Host i915 Scheduling

Dispatch i915 request
Scheduling events

GVT-g workload scheduling

vGPU workload dispatcher
vGPU context switch

vGPU Workload
vGPU context
Scheduling

Scheduling Policy Framework

MMIO Emulation

Virtual HW submission – vExeclist / vGuC

Schedule

Workload Q

intel_vgpu_workload

Workload scheduler
- Pick workload from Q
- Notify virtual HW submission layer
- Shadow context & ring buffer
- Submit shadow context & ring buffer to i915
- Wait workload finished
- Complete workload

Host i915 Submission System
- GVT-g gets the guest frame buffer via virtual display
- VFIO exposes guest frame buffer via dma-buf.
- User application attaches dma-buf into i915 GEM and gets i915 GEM handles
- User application shows guest screen context via i915
Thanks