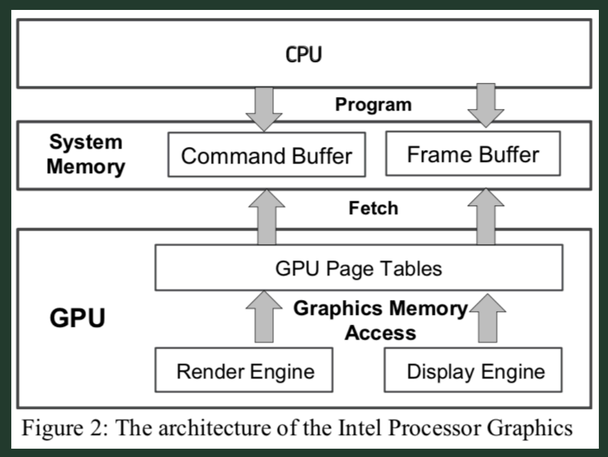

Figure 2: The architecture of the Intel Processor Graphics

Intel Processor Graphics uses system memory as graphics memory

local graphics memory ← only limited to access from the render engine

render engine ---------- fetch GPU commands from command buffer

global graphics memory ---------- mostly the frame buffer, but also serves as the command buffer

dispaly engine ---------- fetch pixel data from the frame buffer and then send them to external monitors for display

the usages of the four critical interfaces

- frame buffer → frequently → so pass-through
- command buffer → frequently → so pass-through
- the GPU Page Table Entries (PTEs) → privileged resources → are trapped and forwarded to the mediator driver in Dom0 for emulation.
- I/O registers → privileged resources → are trapped and forwarded to the mediator driver in Dom0 for emulation.

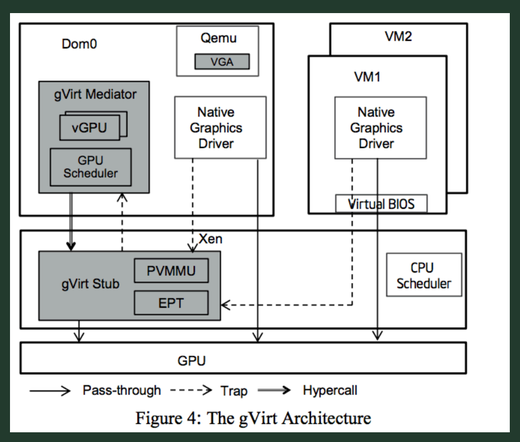

Figure 4: The gVirt Architecture

- Why all VMs share the global graphic memory?
  - ❑ GGTT在硬件中只有一套，如果不让虚拟机复用的话，就需要为每个虚拟机建立一套软件GGTT，每次虚拟机切换时，需要为store与restore虚拟机的GGTT状态，开销太大

gVirt partitions the global graphics memory among VMs.

Graphics memory resource partitioning

- change → Pass-Through → Address space ballooning
- change → Pass-Through → Per-VM local graphics memory

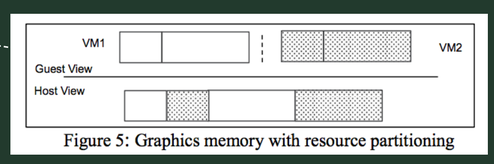

Figure 5: Graphics memory with resource partitioning

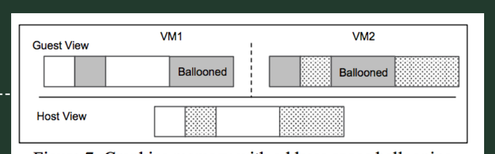

Figure 7: Graphics memory with address space ballooning

why? → driver uses the *local graphics memory* to hold the massive rendering data, while the *global graphics memory* mostly serves only for the frame buffer, and the ring buffer, which are limited in size.

https://www.usenix.org/node/183932
结合视频观看效果更佳！

security

- Inter-VM Isolation
  - Isolation of CPU accesses
  - Isolation of GPU accesses
  - Denial-of-service attack
- Command Protection → full GPU virtualization. To guarantee no unauthorized address reference from the GPU, gVirt audits the guest command buffer at the time of command submission. However there exists a window, between the time when the commands are submitted and when they are actually executed, so a malicious VM may break the isolation by modifying the commands within that window. General → solutions? → smart shadowing mechanism
  - Lazy-shadowing to the ring buffer
  - Write-Protection to the batch buffer