gScale: Improve vGPU Scalability Using Dynamic Resource Sharing

Aug 2016

Pei Zhang Pei.zhang@intel.com
Kevin Tian Kevin.tian@intel.com
Xiao Zheng Xiao.zheng@intel.com
Agenda

- vGPU Scalability of GVT-g
- gScale Design for Doubled vGPU Density
- Evaluation
- Summary
vGPU Scalability in GVT-g
GVT-g Introduction

• Full GPU virtualization solution with mediated pass-through approach

• Open source implementation for Xen/KVM (aka XenGT/KVMGT)
  • Support a rich span of Intel® Processor Graphics
  • Available in https://01.org/igvt-g

Near to native GPU performance + Full GPU capabilities in VM + Flexible VMs sharing (Up to 7VMs)

gScale

>= 2X higher density!

• Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
• For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
Processor Graphics: Components

- CPU
  - Aperture
  - Registers/MMIO
    - Display Engine
    - Render Engine
      - Global Graphics Virtual Memory
      - Per-Process Graphics Virtual Memory
        - GGTT
        - PPGTT
  - System Memory
Shared Global Graphics Memory in GVT-g

- Parallel access from multiple engines due to split vCPU/vGPU scheduling

By guest gfx drivers thru vCPU scheduling

By guest GPU commands thru vGPU scheduling

Upon Foreground/background Display switch
Static Partitioning of Global Graphics Memory

- Minimal 128MB visible/384MB invisible per vGPU
- Means up to 7 vGPUs possible (besides reserved for Dom0)
- Some BIOS may support a smaller aperture which means more limitation
gScale Design for Doubled vGPU Density
gScale: per-VM Global Graphics Memory

Key challenge is to remove parallel accesses!
3 Components in Parallel Access to GGTT

- Render engine access
- Display engine access
- CPU tiled/non-tiled memory access
Render Engine Accesses

- At anytime, only one vGPU’s graphics memory is accessible by render engine. Controlled by vGPU scheduler.
- Dynamically switch per-VM GGTT table at vGPU context switch.
Display Engine Accesses

- Display engine accesses is restricted to the Dom0 reserved region
  Alias mapping of guest framebuffer into reserved graphics memory for Dom0
What is tiled memory

Tiled memory – aperture access
CPU Accesses to Non-Tiled Memory

- **vAperture1**
  - EPT1
- **vAperture2**
  - EPT2

Bypass aperture and directly access system memory.

- Mem access is coherent between CPU cores/GPU.

- **Dom0 reserved region**

- **Global Graphics Memory**
  - per-VM Global GTT

- **VM1 System Memory**
  - Remove use of aperture

- **VM2 System Memory**
CPU Access for Tiled Memory

- vAperture1
- vAperture2
- EPT1
- EPT2
- Tiled Memory
- Aperture
- Fence Registers
- Global Graphics Memory
- per-VM Global GTT
- VM1 System Memory
- VM2 System Memory
- Dom0 reserved region

Dynamic allocation of fence register and aperture for tiled memory!
Summary: Global Graphic memory access

- **CPU**
  - Access Tiled mem: remap to DOM-0 Reserved region

- **GPU**
  - Display Engine
    - Per-VM Global memory
    - GGTT
  - Render Engine
    - Remap to only accesses DOM-0 reserved region

- **System Memory**
  - Access None-Tiled mem: pass through to system memory directly
  - Dom0 reserved region

- **Aperture**
  - Only render owner can access memory
Optimization: Split per-VM GGTT into slots

- per-VM GTT switching is based on slots
- Switching for those only slots that shared between other VMs

**Notes:**

1. vGPU0 (for DOM0) has dedicated mem space slots which won’t be shared by other vGPUs.

Whole per-VM GTT table size is about 2MB
Switching entire GTT table is time consumed.
Context switch performance - Private GTT table copy

1. gScale-Basic is the data without memory slot split.
2. Performance data is sampled with 12 VMs.

Source: USENIX ATC (2015), gScale: Scaling up GPU Virtualization with Dynamic Sharing of Graphics Memory Space
Evaluation
Windows Guest: Scalability 3D performance with 12 VMs could achieve up to 80% of 1 VM.

Linux Guest: Scalability of 3D performance with 15 VMs could achieve up to 80% of 1 VM. 2D performance is better by burn out GPU power.
Summary
Summary

• gScale breaks graphics memory resource limitation by introducing per-VM GGTT design
• gScale doubles vGPU density in Intel®@ GVT-g with good scalability
• Converged performance of 15 vGPUs reaches 96% of native performance

• Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
Resource Links

• Project webpage and release: https://01.org/igvt-g

• Project public papers and document: https://01.org/group/2230/documentation-list

• Intel® IDF: GVT-g in Media Cloud: https://01.org/sites/default/files/documentation/sz15_sfts002_100_engf.pdf


Slide 4 is measured by computing the VM creation number in HSW platform.

Slide 18 is measured by computing the context switch time.

Slide 20 is measured by running different 3D workload and compute the total FPS.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

All test in this document are with same hardware configuration: CPU Intel E3-1285 v3 (4 Cores, 3.6GHz), GPU Intel HD Graphics P4700, Memory 32GB, Storage SAMSUNG 850Pro 256GB*3. Test is made by Xue, Mochi.

For more information go to http://www.intel.com/performance.

Intel, the Intel logo Intel® are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© 2016 Intel Corporation.