

Graphics Pass-through with VT-d

Nov-19-2009

**Weidong Han
Ben Lin**

Xen Summit Asia 2009

Agenda

- Graphics Virtualization Introduction
- Graphics Pass-through with VT-d
- Performance
- Conclusion

Requirements on Graphics Virtualization

- Require graphics capabilities in virtualization environment
 - 3D games
 - High definition video playback
 - CAD

Virtual Machine



Approaches

- Device emulation

Emulate a graphics device

- Sharing amongst guests
- High overhead in emulation

- Graphics API remoting

Forward graphics API calls from guest to the external graphics stack via remote procedure call

- Hardware independent
- Need additional software stacks in guest

- Pass-through

Direct access graphics device in guest

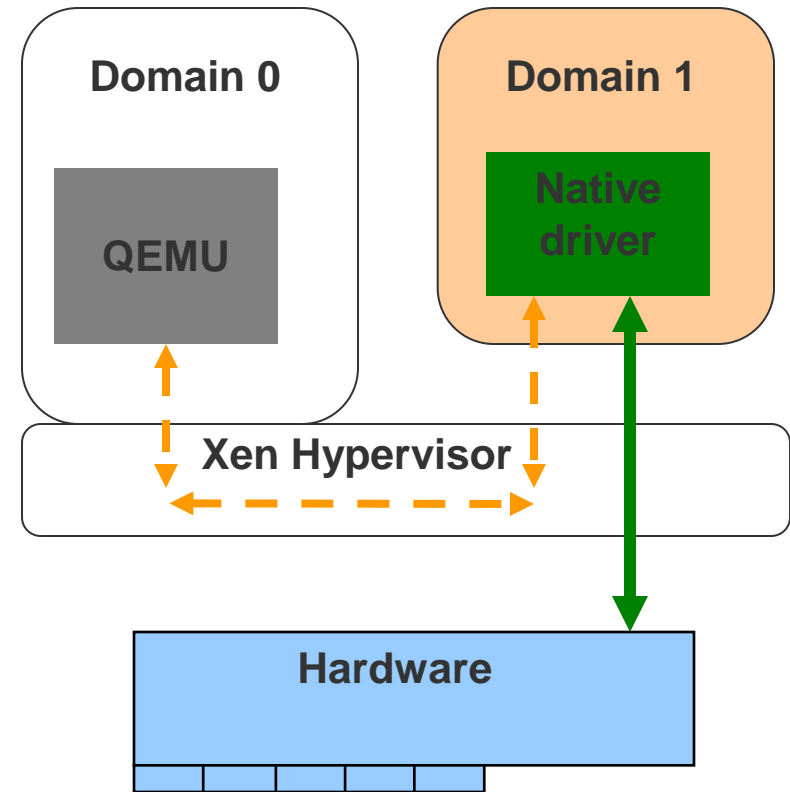
- Good performance
- Don't require any special knowledge of GPU programming interfaces
- Dedicate the graphics device to one guest

Agenda

- Graphics Virtualization Introduction
- **Graphics Pass-through with VT-d**
- Performance
- Conclusion

PCI Pass-through with VT-d

- Assign a physical device to a guest directly
- Guest accesses the device directly with minimal hypervisor intervention
- Physical device accesses guest's memory directly with help of VT-d
- Interrupt will be injected to guest through hypervisor
- PCI Configuration Space Access needs trap-and-emulation



Challenges of Graphics Pass-through

- Legacy VGA resources
 - Legacy VGA I/O ports and frame buffers are at fixed locations
 - I/O ports: 0x3B0 – 0x3BB, 0x3C0 – 0x3DF
 - Frame buffer: 0xA0000 – 0xBFFFF
- Graphics device re-initialization in guest
 - Execute Video BIOS
 - But Video BIOS bits may be truncated or modified after initialization in host, thus cannot be re-executed correctly
- Secondary graphics pass-through
 - The secondary graphics device doesn't claim VGA cycle
- Additional challenges on integrated graphics device (IGD)
 - May have assumptions on host chipset or BIOS

Primary Discrete Graphics Device

- Legacy VGA resources
 - 1:1 map legacy VGA resources

- Video BIOS

For Video BIOS without re-execution issue:

- Extract video BIOS from host address 0xC0000
- Copy video BIOS to guest address 0xC0000

For Video BIOS with re-execution issue:

- Get video BIOS from graphics device via tools (e.g. NiBiTor)
- Load video BIOS file to guest ROMs
- Copy video BIOS to guest address 0xC0000

Secondary Discrete Graphics Device

- Legacy VGA resources
 - 1:1 map legacy VGA resources
- Video BIOS
 - Get video BIOS from graphics device via tools (e.g. NiBiTor)
 - Load video BIOS file to guest ROMs
 - Copy video BIOS to guest address 0xC0000
- Claim VGA cycle
 - Enable VGA on its PCI bridge

Intel Integrated Graphics Device

- Extra changes required to handle assumptions on host chipset or BIOS

- Intel Q45 IGD:

- Assumptions on chipset

- Graphics Control Register: 0x52 – 0x53
 - Specify the amount of graphics memory allocated from system memory
 - Vendor ID and Device ID
 - Linux loads its driver only if it's a compatible chipset

Solution: Pass-through some registers of host bridge

- Assumptions on device BDF (Bus:Device.Function)

- Video BIOS hardcode the BDF to 00:02.0

Solution: Assign BDF 00:02.0 to IGD in guest

- Vendor specific FLR (Function Level Reset)

- FLR capability is not exposed,

Solution: need specific logic to locate it and perform FLR

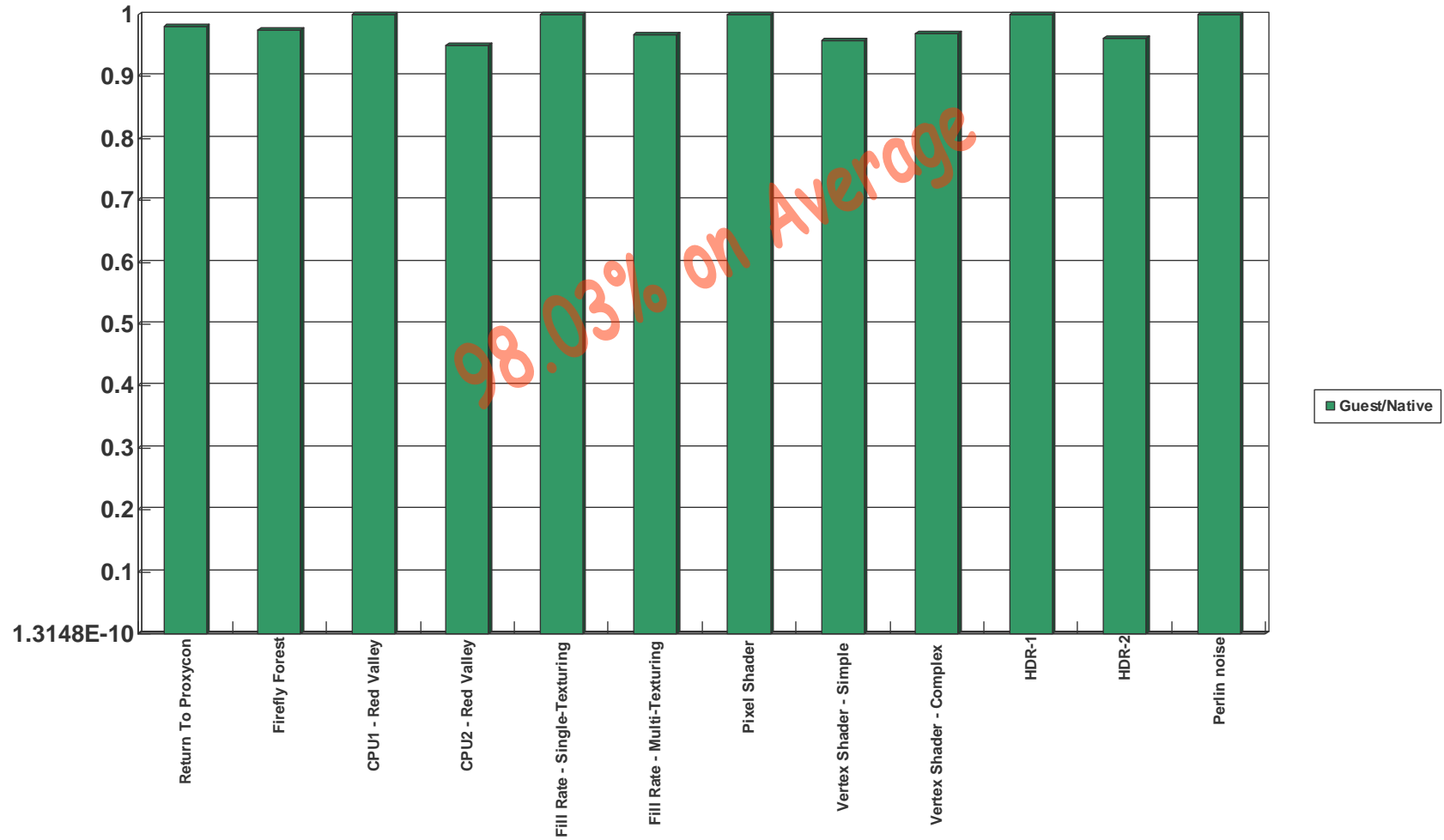
Agenda

- Graphics Virtualization Introduction
- Graphics Pass-through with VT-d
- **Performance**
- Conclusion

Test Environment

- GPU: Intel Integrated Graphics on Intel Q45 Chipset
- Native
 - One CPU
 - Memory 1G
- Guest
 - One Virtual CPU
 - Memory 1G
- Operating System
 - Vista Enterprise 32-bit
 - WDDM driver
- Xen 3.4
- 3DMark06

Results



Agenda

- Graphics Virtualization Introduction
- Graphics Pass-through with VT-d
- Performance
- **Conclusion**

Conclusion

- Graphics pass-through has special challenges
- Xen now supports graphics pass-through:
 - IGD pass-through patch was sent out in xen-devel mailing list, still under review
- Graphics pass-through achieves good performance

Questions?

Backup

Specific Hacks for Graphics Pass-through

- Some graphics devices are virtualization unfriendly
- 1:1 map physical BARs and virtual BARs
 - Some discrete graphics devices (e.g. nVidia Geforce 8400 GS) seem to use the BAR registers in hardware for address decoding of memory requests
- Don't remove shadows of MMIO direct
 - When pass-through some graphics devices to Vista or Win7 guest, it will remove shadows of MMIO direct, then results in system crash

Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All dates provided are subject to change without notice.

Intel is a trademark of Intel Corporation in the U.S. and other countries.

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

Copyright © 2007, Intel Corporation. All rights are protected.

