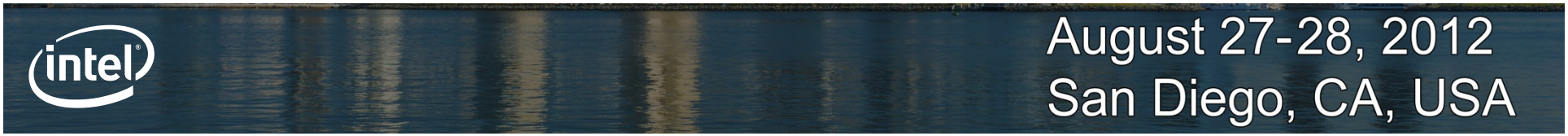
Intel Update

Jun Nakajima

Intel Corporation



Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. 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 PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

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

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2012 Intel Corporation.

Agenda

- Useful VMX features and Example Usages for Xen
  - Descriptor-Table Exiting
  - Accessed and Dirty Flags for EPT
  - VMFUNC
- New Features for Interrupt/APIC Virtualization

Descriptor-Table Exiting

- **Access to GDTR or IDTR. Guest software attempted to execute LGDT, LIDT, SGDT, or SIDT**
  - VM Exit Reason: 46
- **Access to LDTR or TR. Guest software attempted to execute LLDT, LTR, SLDT, or STR**
  - VM Exit Reason: 47

Descriptor-Table Exiting: Example Usages

- **Detect unusual activities with GDT/IDT in hypervisor**
  - Hooking GDT/IDT is classical malware attack
  - Since virtual platforms can have identical configurations, it's easier to detect such suspicious activities
  - To detect modifications to GDT or IDT entries, hypervisor needs to write-protect GDT and IDT
- **Report incidents to user dashboard, management tool in Cloud**
  - Add API for such monitoring

intel

Accessed and Dirty Flags for EPT

- **Accessed Flag**
  - Processor sets flag whenever it uses EPT paging-structure entry as part of the guest-physical-address translation
- **Dirty Flag**
  - Processor sets flag whenever there is a write to guest-physical address
- **Available in EPT paging structures**
  - A Flag is available for region or page (1GB, 2MB, 4KB)
  - D Flag is available only for page (1GB, 2MB, 4KB)

intel

Accessed and Dirty Flags for EPT: Example Usages

- **Guest memory paging**
  - Xenpaging
- **Fast VM check-pointing for fault tolerance**
  - Log-dirty mode in Xen requires write-protection & EPT violation
  - Use D bits to identify dirty pages
    - Additional software-based optimizations should be helpful
- **Monitor memory activities of VMs in Cloud**
  - Use A/D bits to monitor memory activities with very low overheads
    - Scan EPT page tables every one minute (or user metric)
    - Report # of pages accessed/modified to user dashboard

intel

VMFUNC Instruction

- **Allows code in guest to invoke VM function**
  - Configured by software (such as hypervisor) in VMX root operation.
  - No VM exits (if successful)
- **VM function 0: EPTP switching VMFUNC**
  - Allows code in guest to load new value for EPT pointer (EPTP)
  - Loads EPTP from EPTP list (indexed by value of ECX)
  - Does not modify state of any registers; no flags are modified.

VMFUNC Instruction: Example Usages

- **Allow HVM guests to share pages/info with hypervisor in secure fashion**
  1. Hypervisor sets up EPT page tables with additional mapping
  2. VCPU executes VMFUNC instruction in “special” thread
  3. Upon successful execution, it can access additional space that other VCPUs cannot access
  4. Hypervisor forces VCPU to use usual ETP page tables after job is done
- **Optimizations for grant page tables**
  1. Hypervisor sets up EPT page tables so that front-end and back-end can share pages (buffers)
  2. VCPU granted executes VMFUNC instruction to share buffers only for data transfer between front-end and back-end

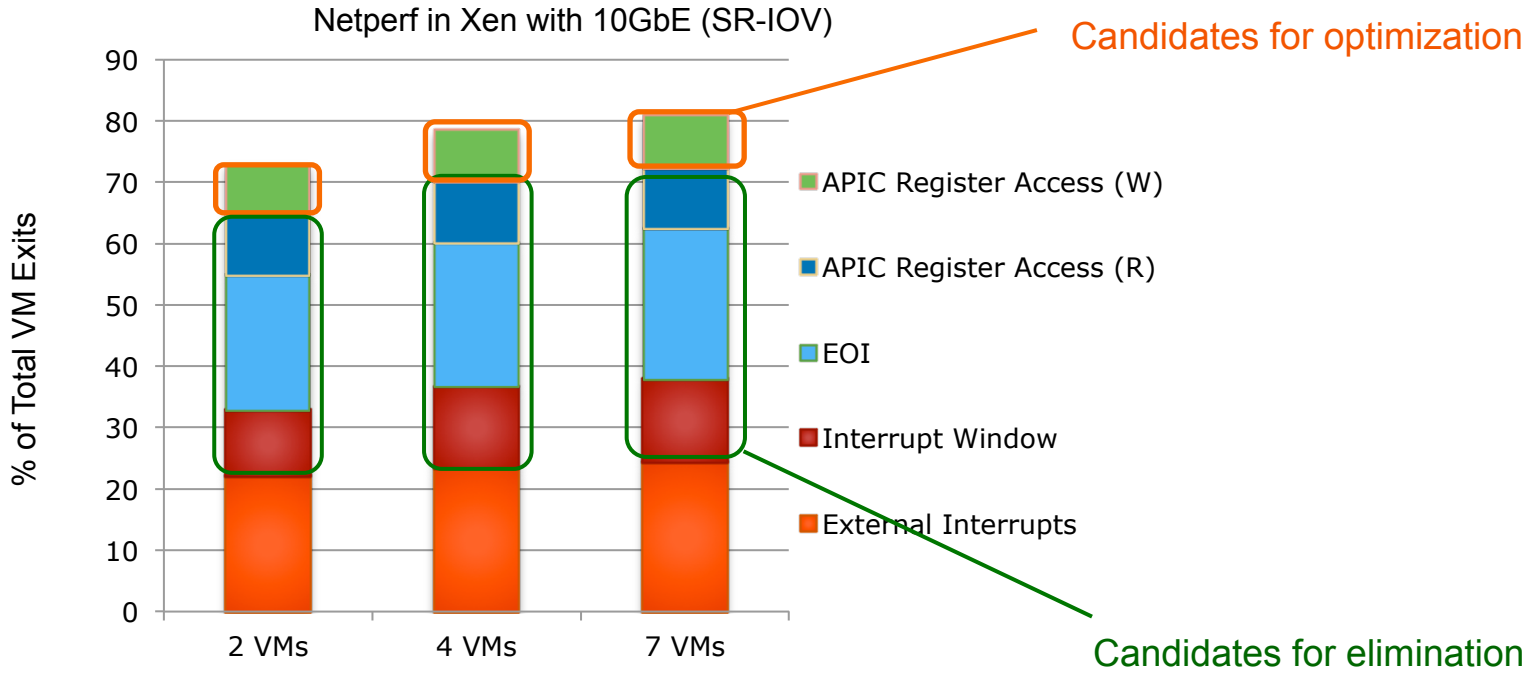
Agenda

- Useful VMX Features and Example Usages for Xen
  - Descriptor-Table Exiting
  - Accessed and Dirty Flags for EPT
  - VMFUNC
- New Features for Interrupt/APIC Virtualization

Interrupt/APIC Virtualization: Overview

- **VMM must virtualize guest's interrupts and interrupt controller (APIC)**
  - Models APIC control state on a "virtual-APIC page" in memory
- **VMM must emulate nearly all guest accesses to APIC control registers**
  - Requires "VM exits" – time-consuming transitions into VMM for emulation – and back
  - VMM must decode and emulate guest instructions that access APIC
  - Except for Intel® VT FlexPriority, which virtualizes access to one APIC control register
    - Task priority – TPR
    - No VM exits required in this case
- **VMM must virtualize all interrupts coming to guest**
  - Must determine when guest is ready to receive interrupts and deliver as needed
- **Virtualization of interrupts and APIC is a major source of overhead**
  - Illustration on next slide

Interrupt/APIC Virtualization: Major Source of Overhead*



- **Performance cost of virtualization mostly due to VM exits**
- **Significant fraction of VM exits are for APIC & interrupt virtualization**
- **Opportunities:**
  - Eliminate entirely VM exits for operations that can be performed by CPU
  - Optimize handling of remaining VM exits by simplifying task of emulation

*Intel internal estimation

Motivations for Further Optimizations

- **Reduce unique overheads of virtualization**
  - Intel is fanatically committed
- **Virtualization has come to be default deployment platform for IT**
  - Any application, even most performance demanding, may run in virtualization
- **Virtualization is foundation of Cloud**
  - More I/O performance/scalability for Web apps, Database, Big Data, HPC, etc.

New Features for Interrupt/APIC Virtualization

- **APIC-register virtualization:**
  - Redirects most guest APIC reads/writes to virtual-APIC page
  - Most reads will be allowed without VM exits
    - such as, interrupt command register - ICR_Low
  - VM exits occur **after** writes (no need for decode)
    - such as, ICR_low, timer's initial-count register
- **Virtual-interrupt delivery:**
  - Extend TPR virtualization to other APIC registers
  - No need for VM exits for most frequent accesses (e.g., EOI – required for every interrupt)
  - CPU delivers virtual interrupts to guest (including virtual IPIs)
  - VMM needn't track guest readiness or deliver manually
    - Eliminates old "pending interrupt" VM exits
- **Net result*: (Intel Netperf estimation)**
  - Eliminate up to 50% of VM exits (most of those related to virtualization of interrupts/APIC)
  - Optimize up to 10% of VM exits (emulation made easier for some APIC writes)

*Intel internal estimation

intel

Call To Action

- **Use existing VMX features to enhance Xen:**
  - Descriptor-Table Exiting
- **Get ready. Spec is already in Software Developer's Manual*:**
  - Accessed and Dirty Flags for EPT
  - VMFUNC
- **Stay tuned:**
  - New Features for Interrupt/APIC Virtualization
    - Software Developer's Manual now added the spec (end of August 2012)

*Intel® 64 and IA-32 Architectures Software Developer's Manual , May 2012