

Intel Updates

Jun Nakajima

Intel Open Source Technology Center

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 © 2007 Intel Corporation.

Throughout this presentation:

VT-x refers to Intel® VT for IA-32 and Intel® 64

VT-i refers to the Intel® VT for IA-64, and

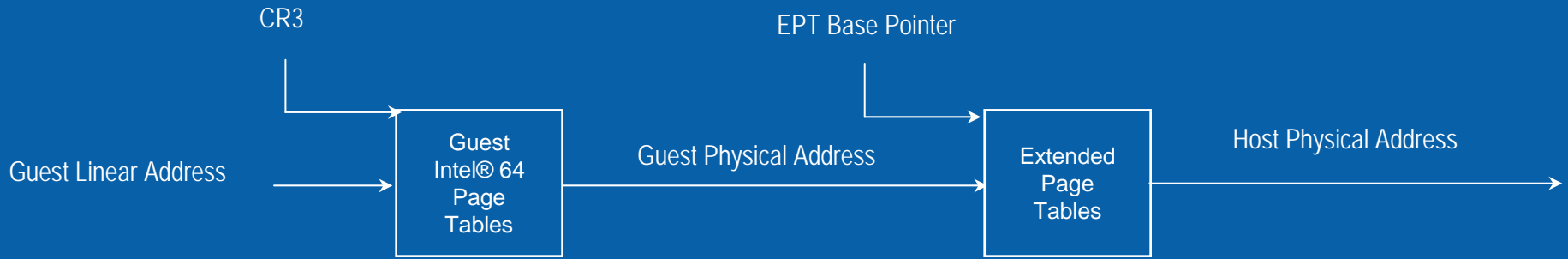
VT-d refers to Intel® VT for Directed I/O



Agenda

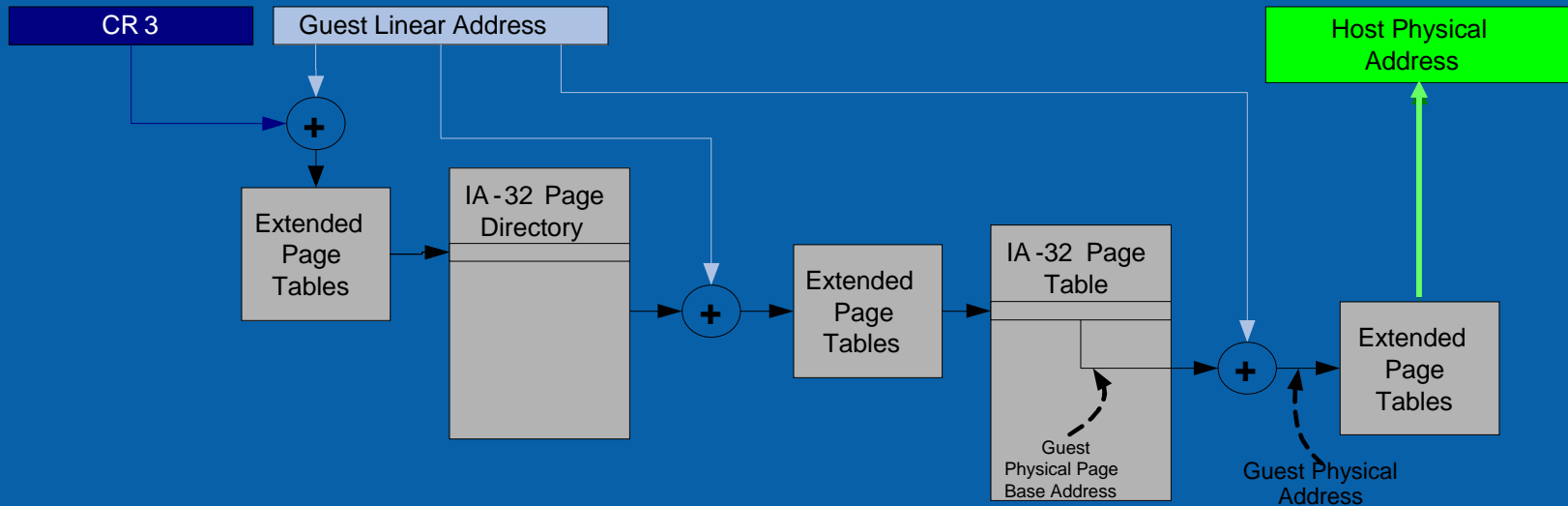
- VT Update
 - EPT and VPID
 - VT-x Microarchitecture Enhancements
- Intel Projects Update
 - VT Enabling
 - Power management in virtualization
 - Hybrid virtualization (hardware-assisted + para)
 - Platform support
 - Intel® Trusted Execution Technology (Intel® TXT)
 - Xen for the Client Environment

EPT: Overview



- Guest can have full control over Intel® 64 page tables / events
 - CR3, CR0, CR4 paging bits, INVLPG, page fault
- VMM controls Extended Page Tables
- CPU uses both tables
- EPT (optionally) activated on VM entry
 - When EPT active, EPT base pointer (loaded on VM entry from VMCS) points to extended page tables
 - EPT deactivated on VM exit

EPT Translation: Details



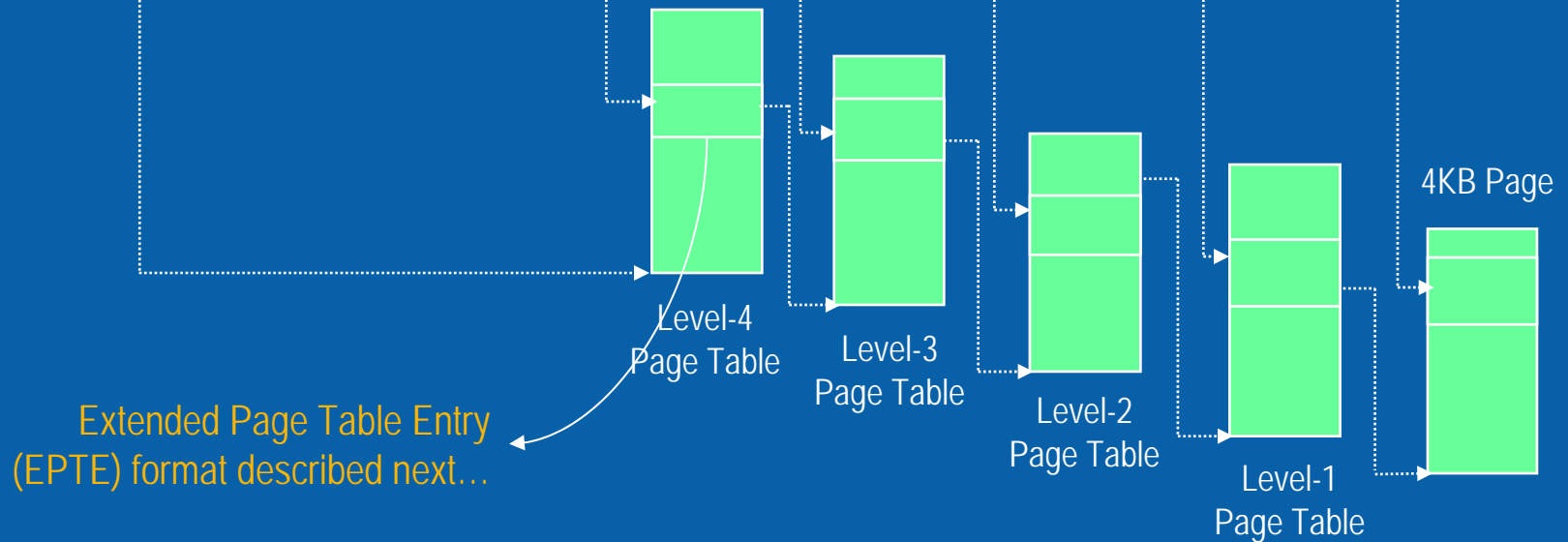
- All guest-physical addresses go through extended page tables
 - Includes address in CR3, address in PDE, address in PTE, etc.
- Example given is for basic 32-bit paging
 - Also applies to other paging modes (e.g., PAE and Intel® 64)
- At leaf, Intel® 64 page faults recognized before EPT violations

Physical Address Translation

Guest Physical Address



EPT Base Pointer



Extended Page Table Entry Format



- ADDR: Physical address of next table (if not super page and not EPTE) or of page frame (if super page or terminal entry)
- SP: Super-page bit: if set, walk stops at a large page
- Permission bits: read (R), write (W), execute (X)
- EPT MT: Memory-typing controls
- AVL: Software-available bits

Caching of EPT-Based Translations

- EPT-based translations used (and cached) only when guest is running
- EPT-based translations may be invalidated only when VMM is running
- EPT-based translations are architecturally tagged with EPT Base Pointer:
 - Tag based only on EPT base pointer
 - Tag not based on guest CR3; guest CR loads still flush guest translations

TLB Management by the VMM: INVEPT

- New instruction to invalidate EPT-based (i.e., “physical”) translations
- Three operands:
 - The flush extent (see below)
 - The 64-bit EPTP indicating the EPT context to be synchronized
 - The 64-bit guest-physical address to be synchronized
- Flush extent operand chooses:
 - Context-wide: invalidation of all translations associated with EPTP operand
 - All-contexts: invalidation of all translations associated with all EPTP values

VPID: New Support for Software Control of TLB

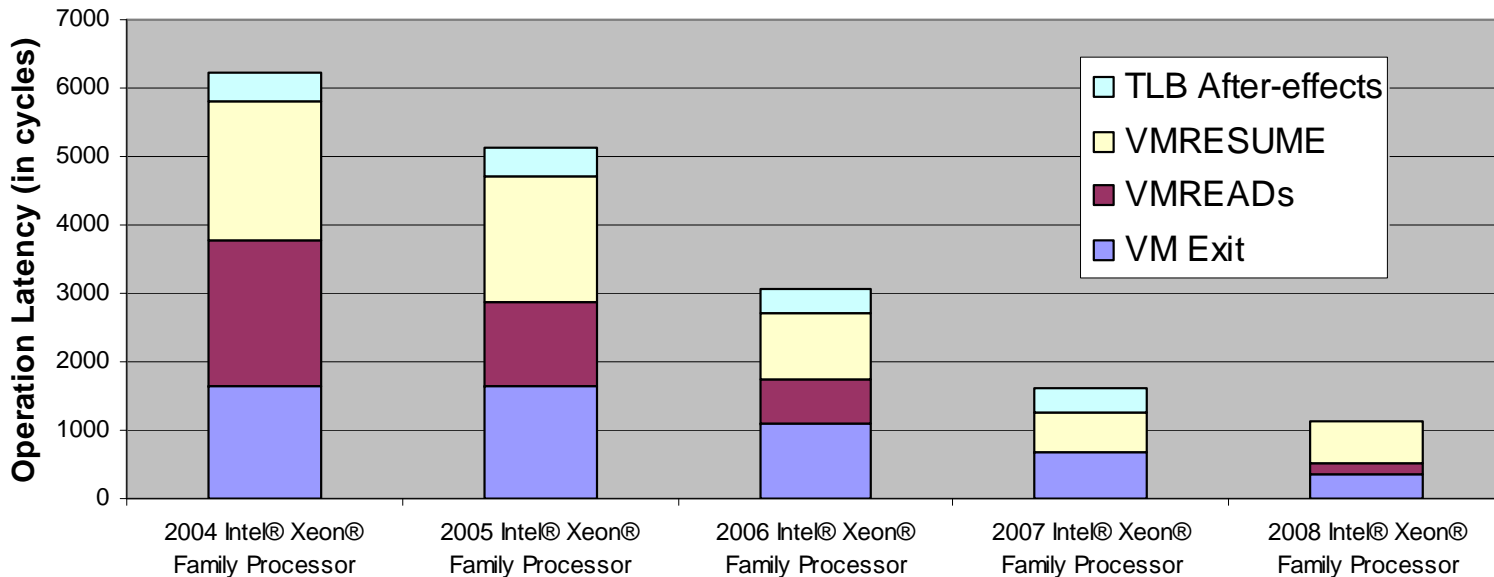
- VPID activated if new “enable VPID” control bit is set in VMCS
- New 16-bit virtual-processor-ID field (VPID) field in VMCS
 - VMM allocates unique value for each guest OS
 - VMM uses VPID of 0x0000, no guest can have this VPID
- Cached linear translations are tagged with VPID value
- No flush of TLBs on VM entry or VM exit if VPID active

TLB Management by the VMM: INVVPID

- New instruction to allow VMM to flush guest mappings
- Three operands:
 - The flush extent (see below)
 - The 16-bit VPID indicating the VPID context to be flushed
 - The 64-bit guest-linear address to be flushed
- Flush extent operand chooses:
 - Address-specific: invalidation of translations associated with VPID and address operands
 - Context-wide: invalidation of all translations associated with VPID operand
 - Context-wide preserving global translations: invalidation of all non-global translations associated with VPID operand
 - All-context: invalidation of all translations associated with all VPID values
- Allows VMM to emulate Intel® 64 paging faithfully

Latency Reductions by CPU Implementation

Intel® VT-x Transition Latencies by CPU



- Further improvements planned for future implementations

***VMX Transition and Instruction Latency
Improvements are dramatic and continuing***

Xen Summit November 2007 †. Measurements based on microbenchmark tests of VM entry / exit times on different Intel® implementations. Actual performance may vary (e.g., based on CPU freq).



Enabling Power Management for Xen

- Cx, Px Support
 - Have a C & P state governor in Xen
 - Port Linux code
 - Parse the ACPI tables in user space
 - Use hypercall to pass the info to Xen
 - Works for non-Linux dom0
 - Deeper Cx support in Xen
 - TSC and local APIC can stop

Xen will have optimal power management for Cx and Px support

Hybrid Virtualization

- Start from hardware-assisted full virtualization
 - Consistent and well-defined CPU behavior
 - Benefit from future silicon enhancements for hardware-assisted virtualization
 - More features, lower VM entry/exits costs
- Use para-virtualization on the focused areas
 - Reduce virtualization overheads
 - Improve cache utilization
 - Simplify the implementation
- Common binary as the native
 - Can be installed for the native and virtualization
- VMM-Agnostic
 - Single para-virtualization code in Linux for various VMMs

VMM-Agnostic Para-Virtualization

- Detected by CPUID (e.g. leaf 0x4000_00xx) on x86/x86-64
 - Never detected by the native
- Pseudo H/W features
 - MMU (e.g. direct paging mode, large pages)
 - I/O
 - Interrupt controllers
 - Time/idle
 - SMP (IPI)
- Allow the same guest Linux binary to use the single para-virtualization code across VMMs
 - Otherwise, detect a VMM, then use the the VMM-specific para-virt code

Hybrid Virtualization – Status

- Ported Xen MMU to KVM
 - Works on KVM
 - Adding other PV
- Rebasing the code using the i386/x86-64 merged tree
- Testing the same binary on Xen

Hybrid Virtualization provides HVM guests with Xen PV interoperability across different VMMs

Summary

- VT Update
 - EPT and VPID
 - VMX Transition and instruction latency improvements are dramatic and continuing
- Xen will have optimal power management for Cx and Px support
- Hybrid virtualization provides HVM guests with Xen PV interoperability across different VMMs

Your participation is very welcome!

