



Xen and Intel® Virtualization Technology for IA-64

YaoZu (Eddie) Dong
Software and Solution Group
Open Source Technology Center



Virtualization

- **Start of day**

- Firstly introduced by IBM in 1960s to share the expansive mainframe system
- Widely extended to PC since later 1990s

- **Why virtualization**

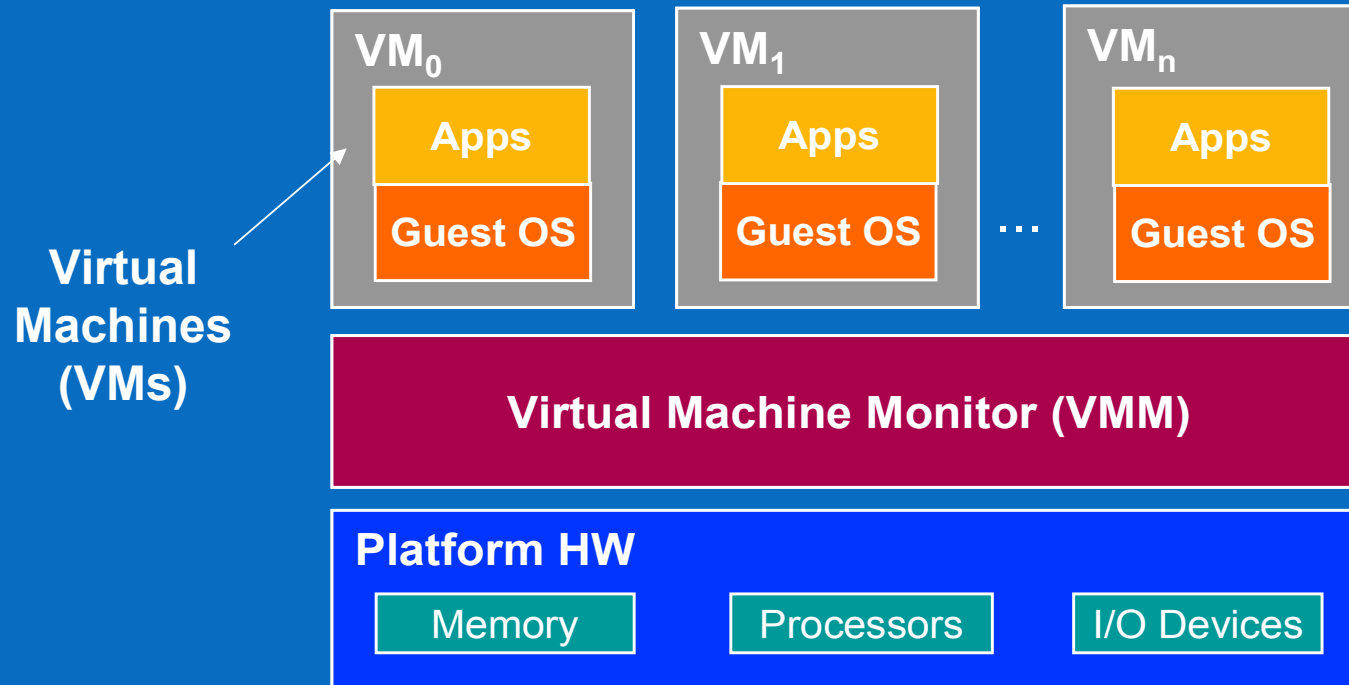
- Allow a single machine to replica itself, AKA Virtual Machines
- Allow each Virtual Machine to run its own operating system
 - Flexible server consolidation
 - Enhanced availability and security
 - Simpler OS and hardware migrations
 - Etc.

- **IA-64 virtualization**

- Itanium is one of major architecture in high end server
- Virtualization support is critical in server and data center



Virtual Machine Monitor



- **VMM transforms the single platform into multiple**
 - Support multiple guest OSes
 - De-privilege each OS to run as Guest OS

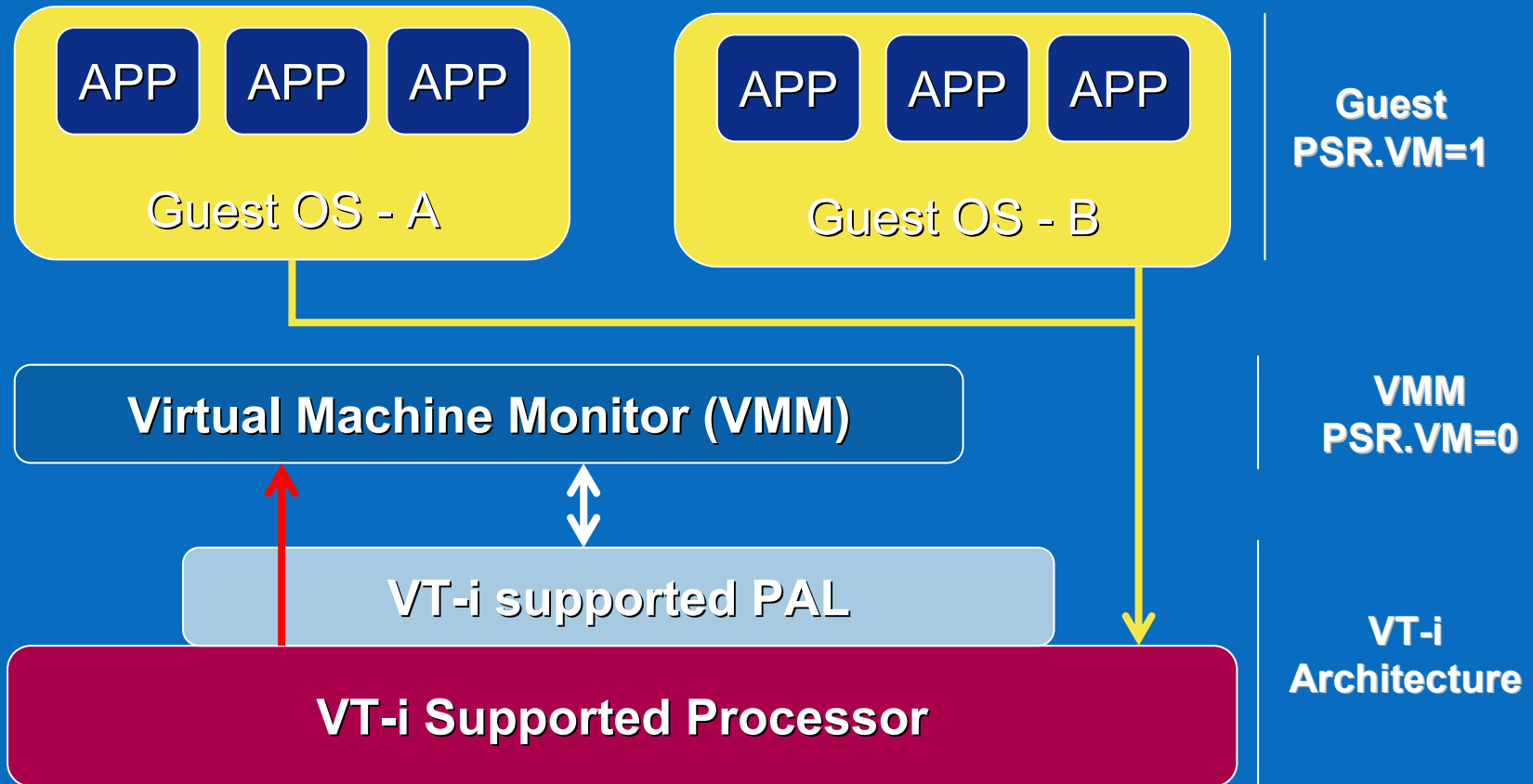
Itanium Virtualization

- **Top IA64 Virtualization Holes**
 - Ring compression
 - Non-trapping instructions
 - Interrupt virtualization issues
 - Address space compression
- **Paravirtualization**
 - Modify guest Operating System to cooperatively work with hypervisor
 - But ...
 - Validation effort & TTM
 - The interface to hypervisor varies from VMM to VMM
 - Difficult to support Proprietary OS such as Windows
- **Intel® VT for IA-64.**
 - Silicon level virtualization support to eliminate virtualization holes and the need for complex software workarounds
 - Support unmodified OS

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



VT-i Processor Virtualization Overview



VT-i provides architectural completeness and robustness



VT-i Support in PAL

Virtual Processor Descriptor

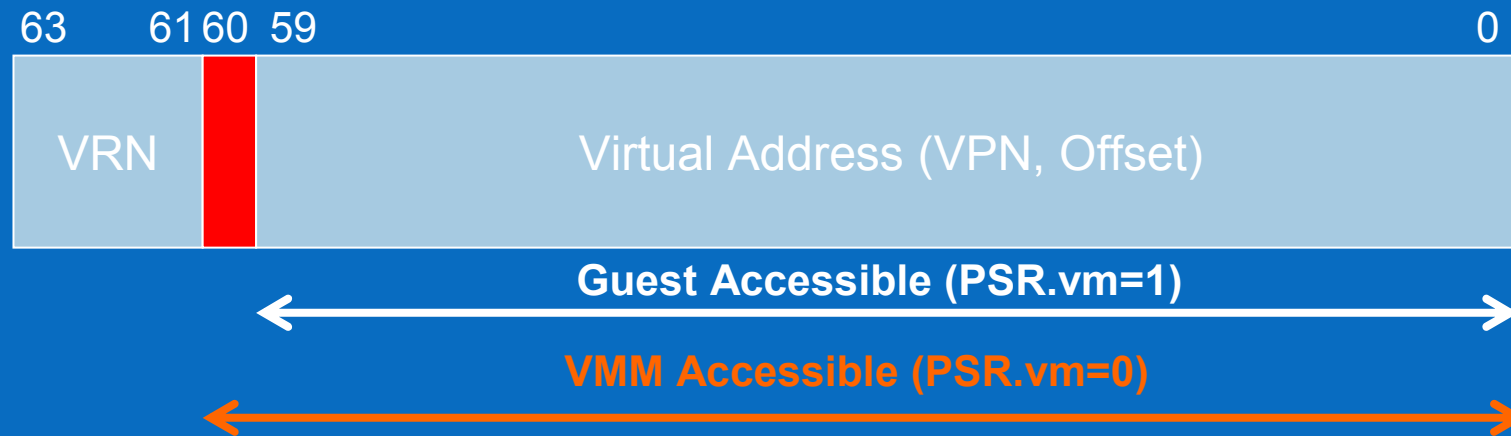
Virtualization Procedures

Virtualization Services

Enhanced VMM Interception

Configuration Options

VT-i: Protect hypervisor from guest address space



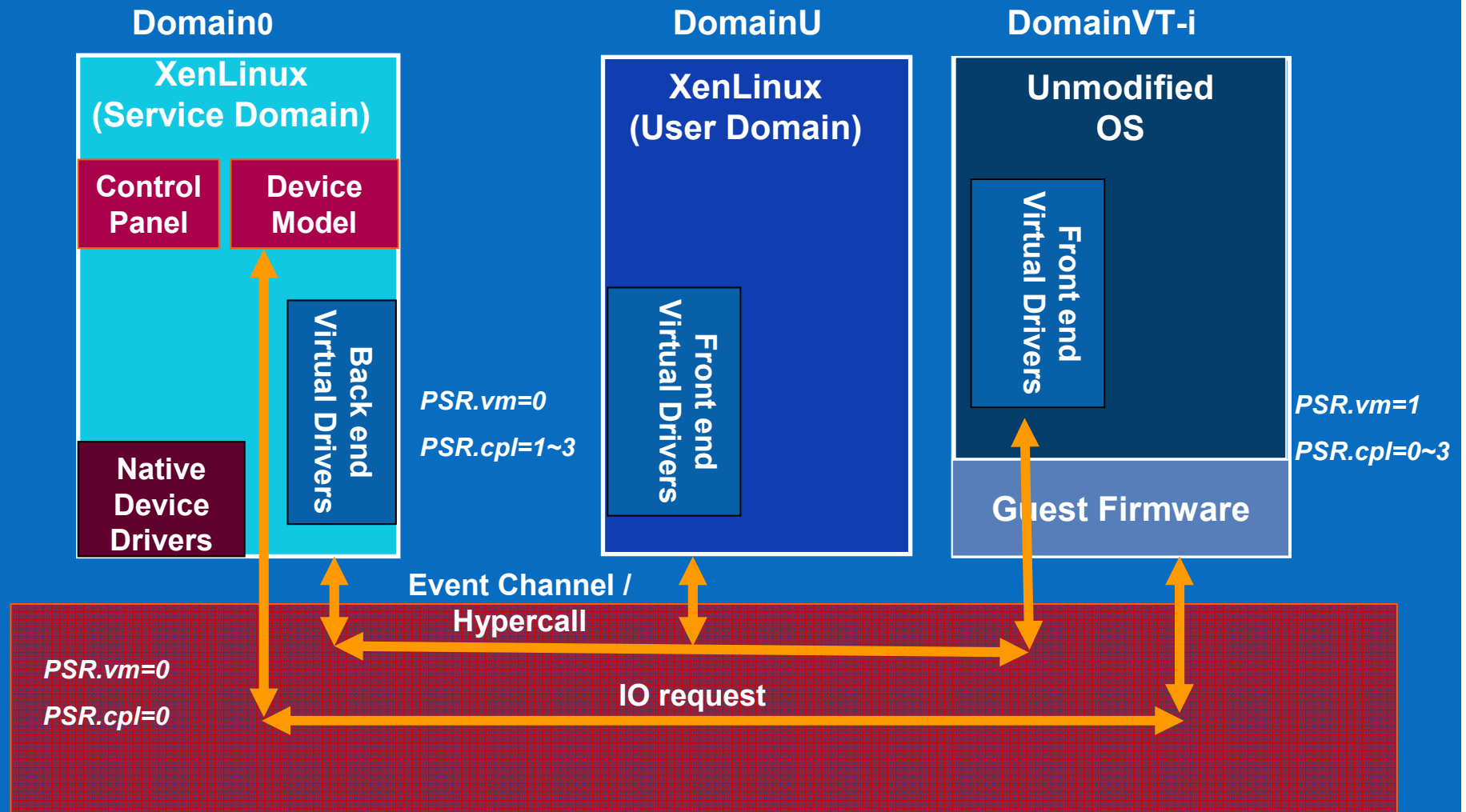
- **Guests see one less address bit**
- **VMM see all supported virtual address bits**

Xen

- **Xen is an open source Virtual Machine Monitor that can boot multiple instances of paravirtualized and unmodified guest OS**
- **Xen is originally developed on X86 architecture**
- **Xen/VT-i: An Hardware Virtual Machine (HVM) Monitor for IA-64**
 - Fully utilize VT-i for architecture completeness and robustness
 - Maximal consistency with IA-32 Xen



Xen with VT-i



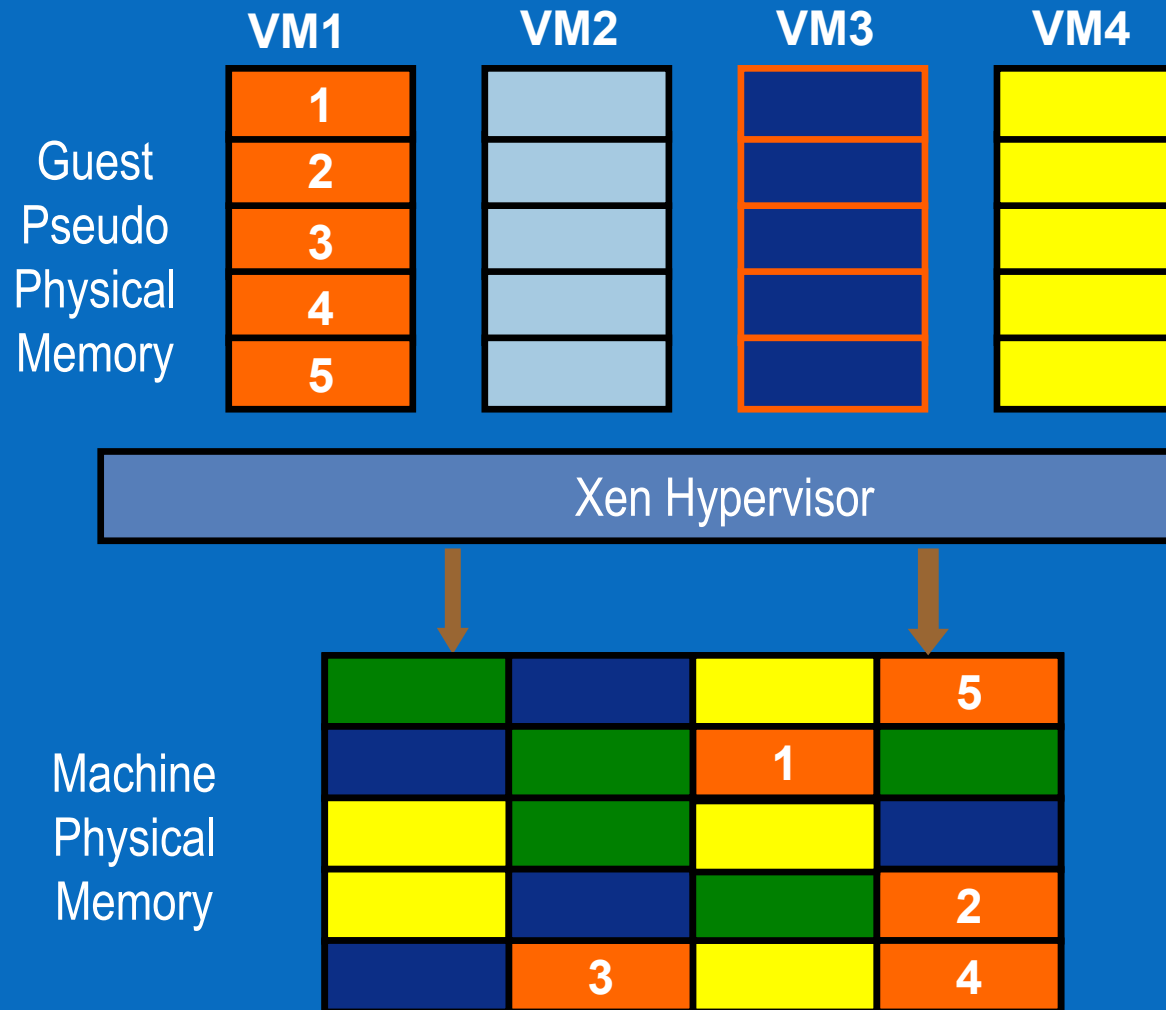
Xen Hypervisor



Memory virtualization



Physical page frame partitioning



Address space - Region ID partitioning

- Simplified TLB format



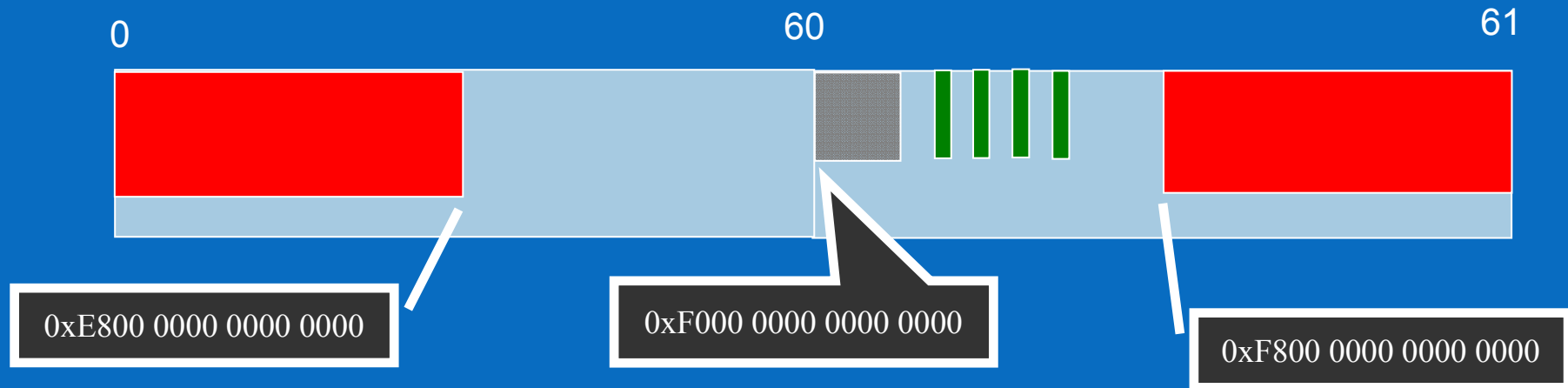
- Partitioned Region ID



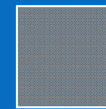
Translation entries from multiple VMs share the same TLB



Xen address space (region 7)



Xen accessible address space



Xen identity map



VT-i domain address space



Xen alias map

Xen hypervisor uses same RID with guest while still gets isolation thru VT-i

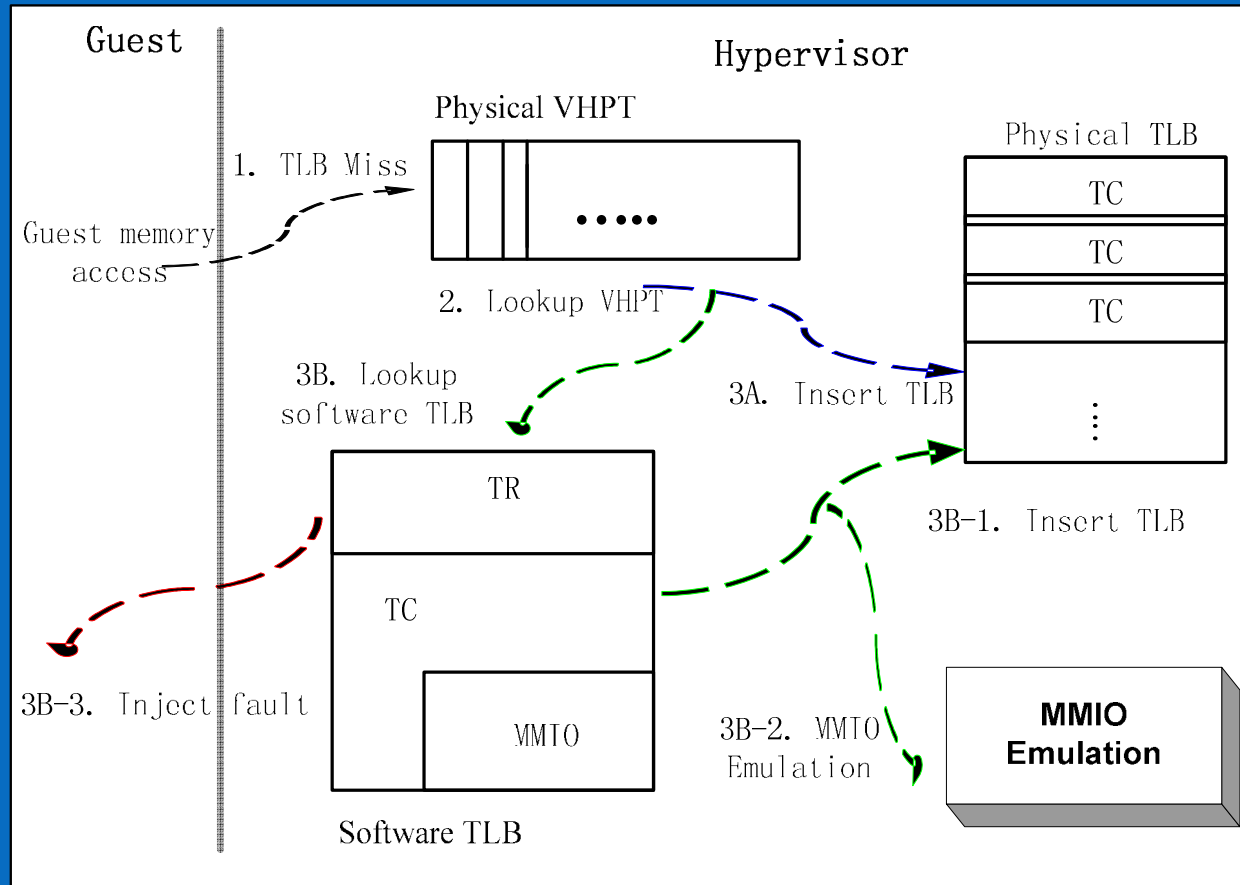
MMU virtualization



Shadow TLB vs. shadow page table

- **Xen/X86 HVM implement shadow page table**
 - Shadow TLB is inefficient in X86
 - Host page fault (VM exit) is very expensive
 - Guest OS purge entire TLBs at process switch time (CR3 write)
 - Excessive page fault will be raised if implementing shadow TLB.
 - Shadow page table
 - Much effective than shadow TLB, but ...
 - Duplicating page table consume both CPU cycles & memory
- **Xen/VT-i HVM implement shadow TLB**
 - Shadow TLB in IA-64 is high efficient
 - IA-64 use RID to differentiate TLBs from different process, thus guest OS rarely flush entire TLBs
 - Guest TLB insert/purge is trapped by VT-i
 - Less complexity and better scalability
 - Natural per VP shadow TLB
 - No guest page table protect issue

TLB miss handling



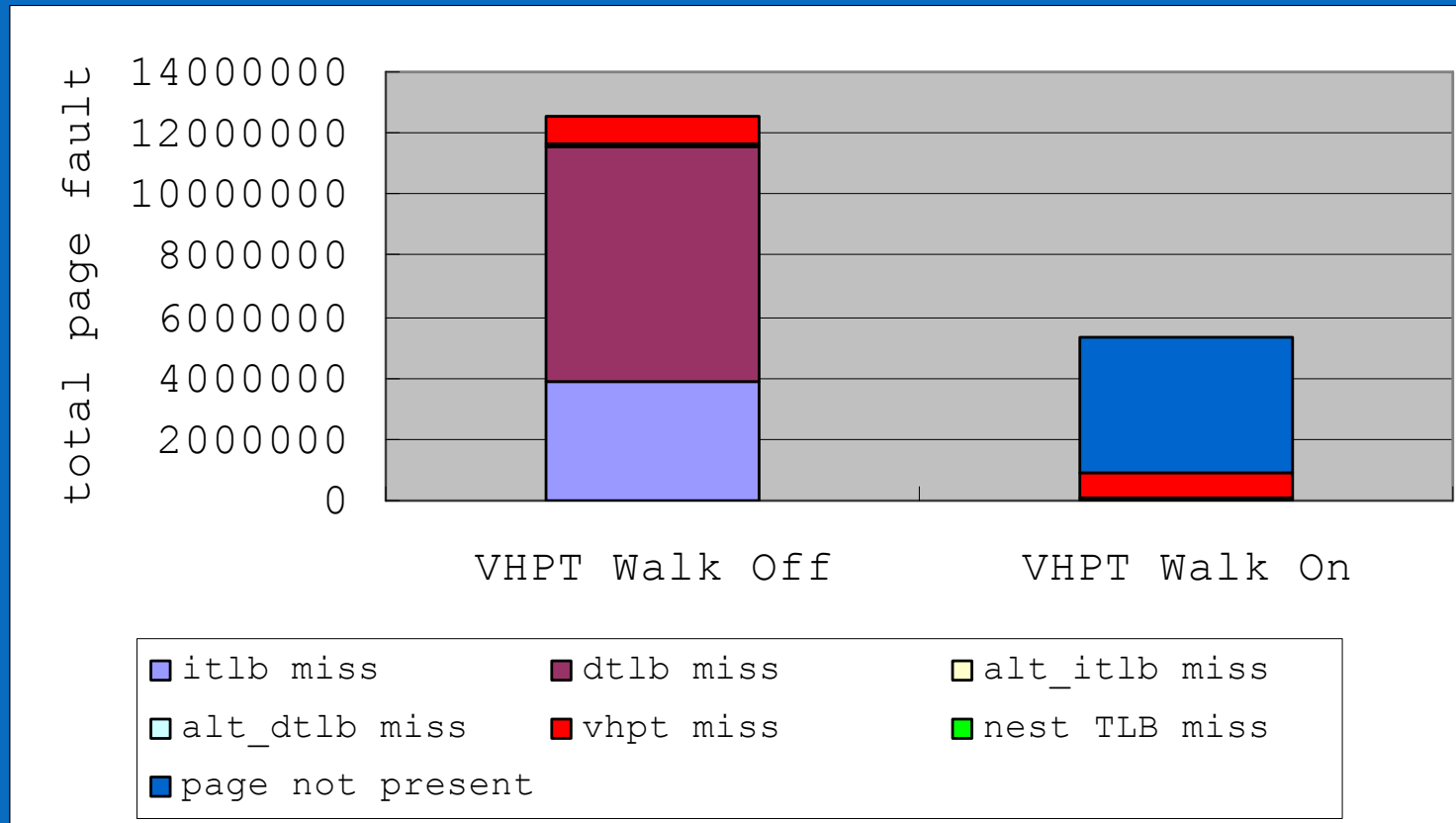
- **Guest TLB resides in physical TLB, physical VHPT and in memory software data structure (swTLB)**

Physical VHPT

- **VHPT could be shared among VMs, but ...**
 - Guest OSes with different page size may co-exist
 - Windows use 8KB page size
 - Linux use 16KB
 - Guest ptc.e emulation
 - Can't distinguish VHPT entries from other VMs
 - Need to invalidate entire VHPT
 - VM performance is interfered each other
 - Scalability
- **Thus ..., Xen/VT-i uses per VP VHPT**

Walking guest VHPT

- **Walking guest VHPT can significantly reduce guest page fault.**
 - Linux Kernel Build performance increases 5.4%
 - Guest page fault is reduced by 58%



Guest TR emulation

- **Use physical TC to emulate guest TR**
 - Guest physical address translated by guest TR can't be guaranteed contiguous or correctly aligned.

- **But ...**

- will guest ld.s to a TR mapped page?
 - Guest may get NAT if exception is deferred
 - Yes for some proprietary OS
 - 0xe000000084005ca0L:
 - ld8.s r28=[r25]
 - nop.m 0x0
 - tnat.nz.or p6,p0=r28

r25=0x1ffffffffc0001be8

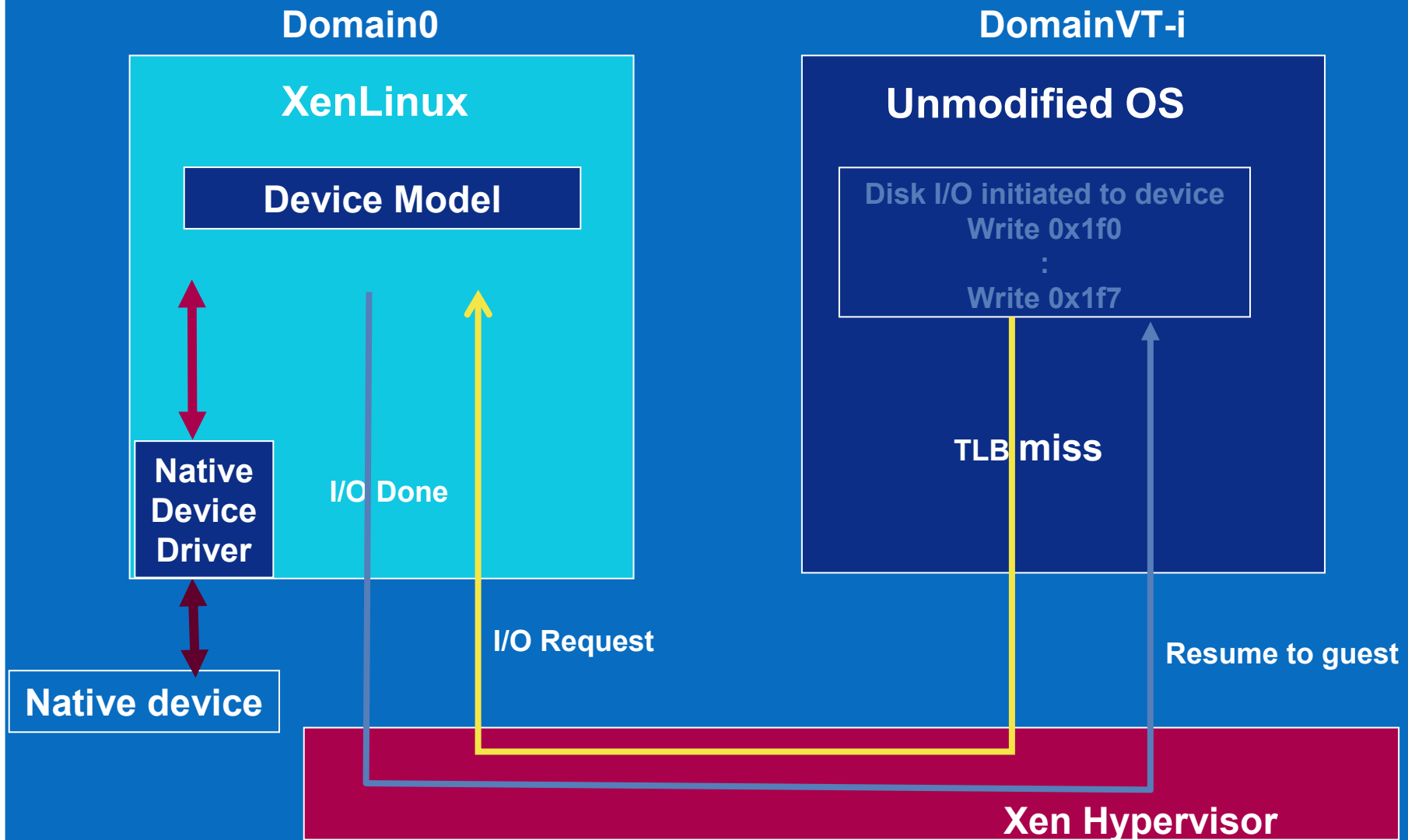
```
Guest DTR[3]:  
000000003c1ee461 0000000000000034  
1ffffffffc000000 00000000000001100  
(va: 0x1ffffffffc000000, size=8K)
```

- **Physical processor need to use more strict exception deferral condition than guest**

IO virtualization



IO emulation



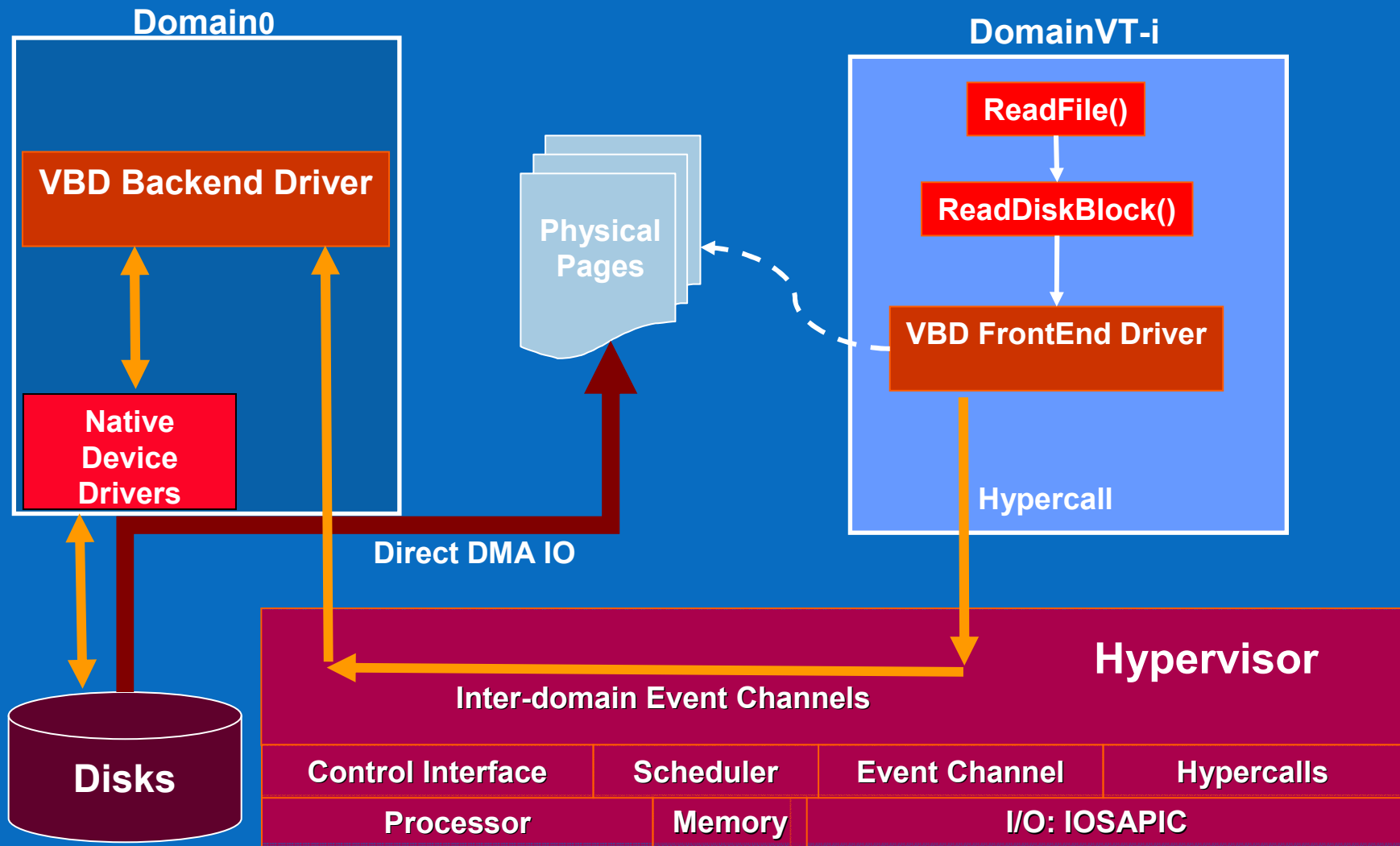
Device Model

- **Reuse X86 device model (Qemu)**
 - Move hot devices to hypervisor
 - LSAPIC/IOSAPIC
 - Buffer I/O write in hypervisor to reduce context switch
 - Standard VGA frame buffer
 - Enhance network device model to be event driven
 - Reduce network package response time and thus throughput.
- **Enable DMA to reduce the excessive I/O data transfer**
 - Block device

Emulation challenge - DMA Cache coherency

- **Native Platform maintain coherency for DMA traffic**
 - Issue snoop cycles on the bus
 - Thus invalidate appropriate I/D cache lines
- **But ... Device Model doesn't guarantee coherency in guest DMA emulation.**
 - Device Model copy data from host DMA buffer to guest DMA buffer
 - I side cache coherency is not guaranteed
 - Remote processor's coherency is not guaranteed
- **Will lose of I side cache coherency matter?**
 - Yes for some proprietary OS
 - OS may execute directly after DMA completion
- **Solution**
 - Synchronize I/D cache and broadcast to remote processors

Virtual Device Driver - VBD



Virtual Device Driver provides High performance IO



Latest status



Latest Xen status

- **Xen 3.0.3 RC1 is published**
 - SMP host & paravirtualized guest support for both X86 & IA-64
 - SMP Linux and Windows support for HVM
 - VBD/VNIF support for both paravirtualized guest & HVM
 - <http://xenbits.xensource.com/xen-3.0.3-testing.hg>
- **Xen/VT-I work in progressing**
 - Performance tuning
 - Stability & Scalability
 - Save/restore and live migration



