Virtualization with Xen and Linux

Chris Wright

chrisw@redhat.com

OSDL-Japan Linux Symposium

June 2006

Outline

- Virtualization Overview
- Xen Architecture
- Xen Current Status
- XenLinux upstream merge
- Xen Roadmap

Note: Much of the information in this presentation comes from papers, web pages and slides found at http://www.cl.cam.ac.uk/Research/SRG/netos/xen/

Virtualization: Why?

- Server consolidation
  - Control physical server proliferation
- Fast and easy provisioning
  - Provision and deploy virtual machine is agile
- Hardware enablement
- Secure isolation
- Test and Debug

Virtualization: History

- Long history
  - 1960's IBM TSS research...1972 S/370 (VM/370)...present S/390
  - 1972, Robert Goldberg 'Architectural Principles for Virtual Computer Systems.' Seminal work describing esp. hardware requirements for virtual machine.
- Virtual Machine
  - Statistically significant number of instructions run on bare machine
  - Sensitive instructions trapped to VMM
    - Real challenge for x86 architecture ;-)
  - Non-privileged instruction symmetry
  - Memory protection

Virtualization Overview

- Partitioning single OS image: Linux-Vservers, OpenVZ, Solaris Zones
  - Group user processes into resource containers
  - Hard to get strong isolation
  - Sensitive to QoS Crosstalk
- Full platform virtualization/emulation: VMware, VirtualPC, QEMU
  - Run multiple unmodified guest OSes
  - Hard to efficiently virtualize x86
- Para-virtualization: UML, Xen
  - Run multiple guest OSes ported to special arch
  - `arch/i386/mach-xen` is very close to normal x86

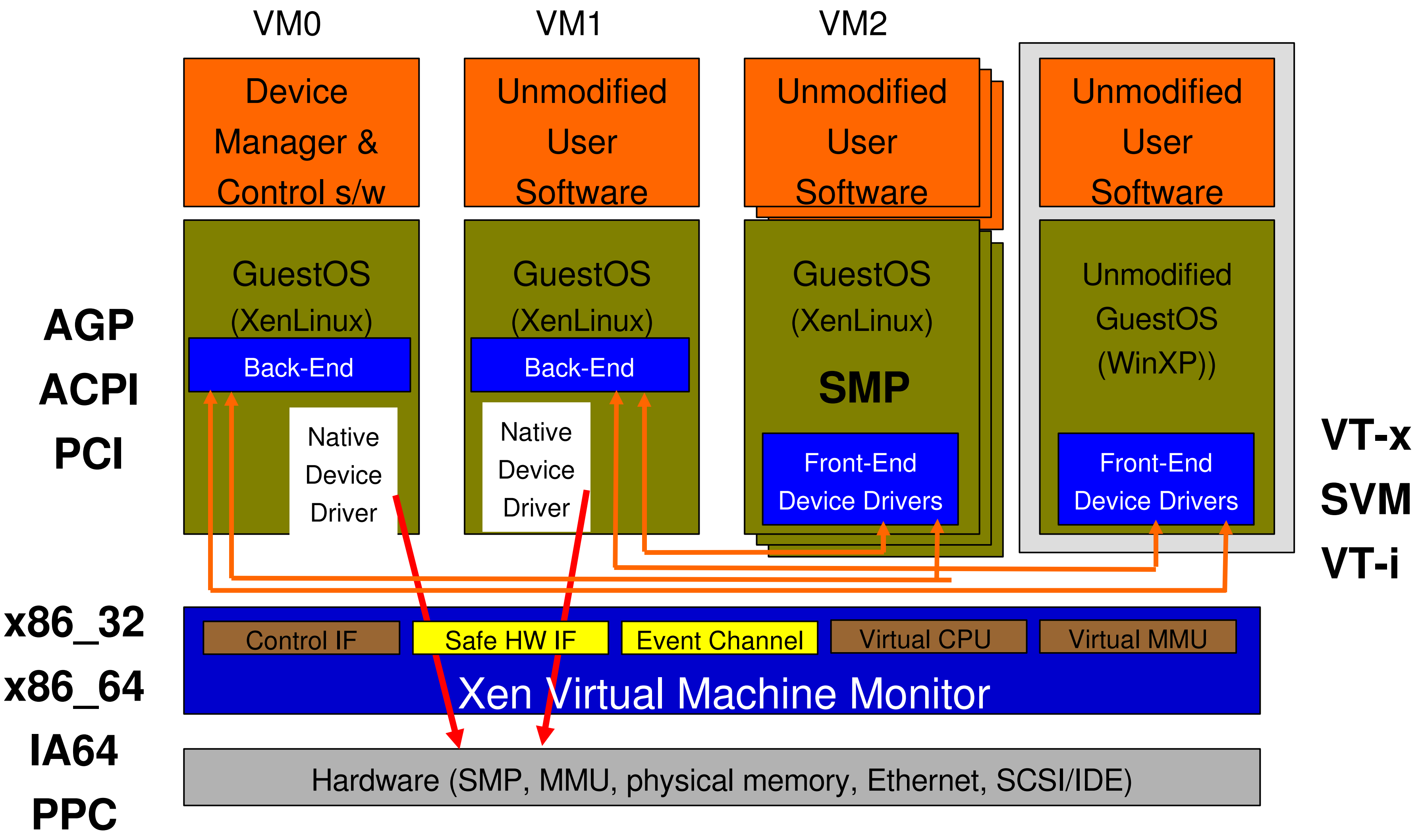
Xen Today: Xen 3.0

- Secure isolation between VMs
- Resource control and QoS
- Prolific guest support
  - Linux, FreeBSD, Solaris, NetBSD, Plan9, Netware
  - Both UP and SMP guests supported
- Execution performance close to native
- Rich hardware support
  - Direct device access (leverage existing driver support)
  - paravirtual i386, x86_64, ia64, PPC, (rumor of SPARC port being underway)
- Support for hardware assisted full virtualization: HVM (VT-x and SVM), VT-i
- Loadable MAC security policy for hypervisor: Chinese Wall, Type Enforcement
- Live migration of VMs

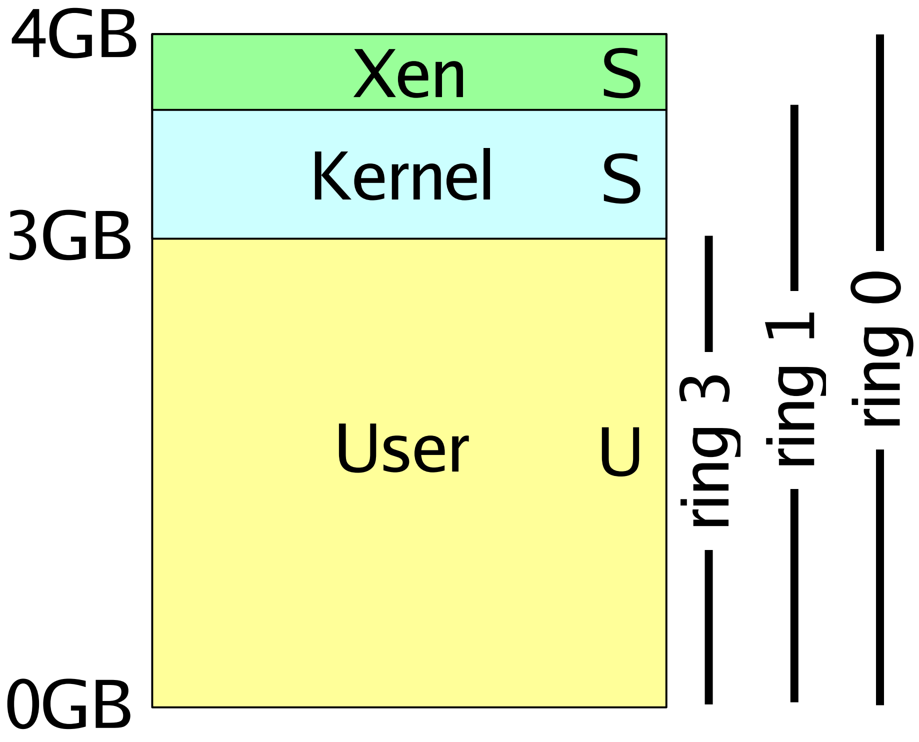
Para-Virtualization in Xen

- Xen provides a new architecture which is very similar to x86
  - Privileged instructions are ported to Xen
    - e.g. LIDT, HLT, load and store CR/DR, INVLPG, CLI/STI
  - Avoids binary rewriting
  - Minimize number of privilege transitions into Xen
    - Shared data structures: read CR2, CLI/STI
    - Batched operations: bulk mmu updates
  - Modifications to Linux are relatively simple and self-contained
- Modify kernel to understand virtualized env.
  - Wall-clock time vs. virtual processor time
    - Xen provides both types of alarm timer
  - Expose real resource availability
    - Enables OS to optimise its own behaviour

Xen 3.0 Architecture

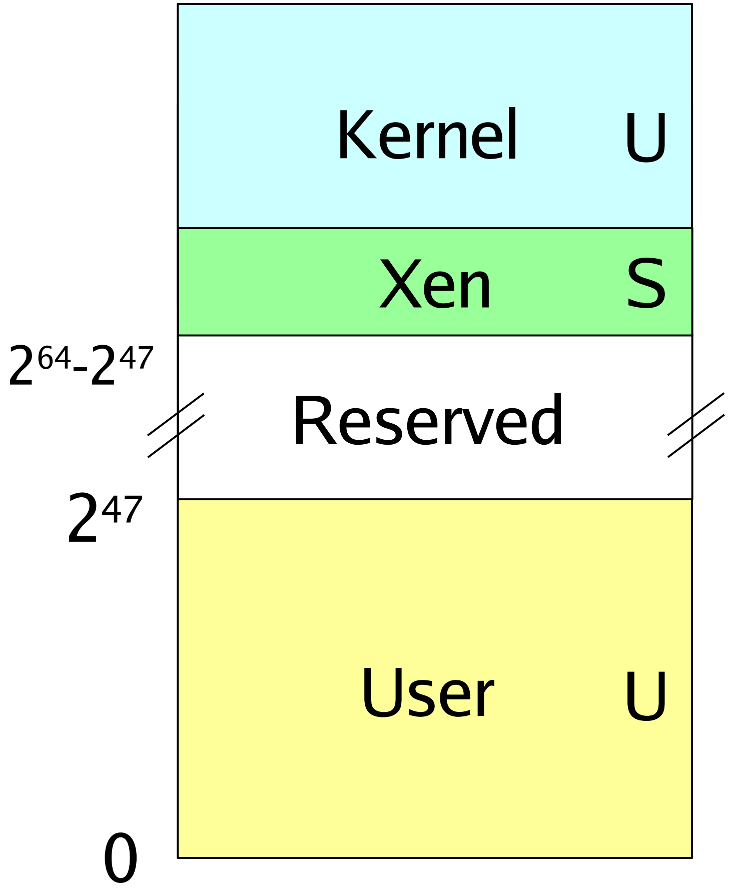


Protection: x86_32



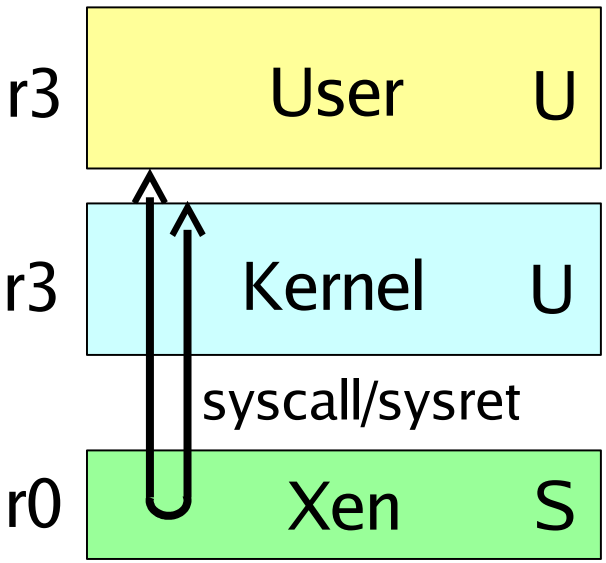
- Xen reserves top of VA space
- Segmentation protects Xen from kernel
- System call speed unchanged
- Xen 3 now supports PAE for >4GB mem

Protection: x86_64



- Large VA space makes life a lot easier, but:
- No segment limit support
- Need to use page-level protection to protect hypervisor

Protection: x86_64



- Run user-space and kernel in ring 3 using different pagetables
  - Two PGD’s (PML4’s): one with user entries; one with user plus kernel entries
- System calls require an additional syscall/sysret via Xen
- Per-CPU trampoline to avoid needing GS in Xen

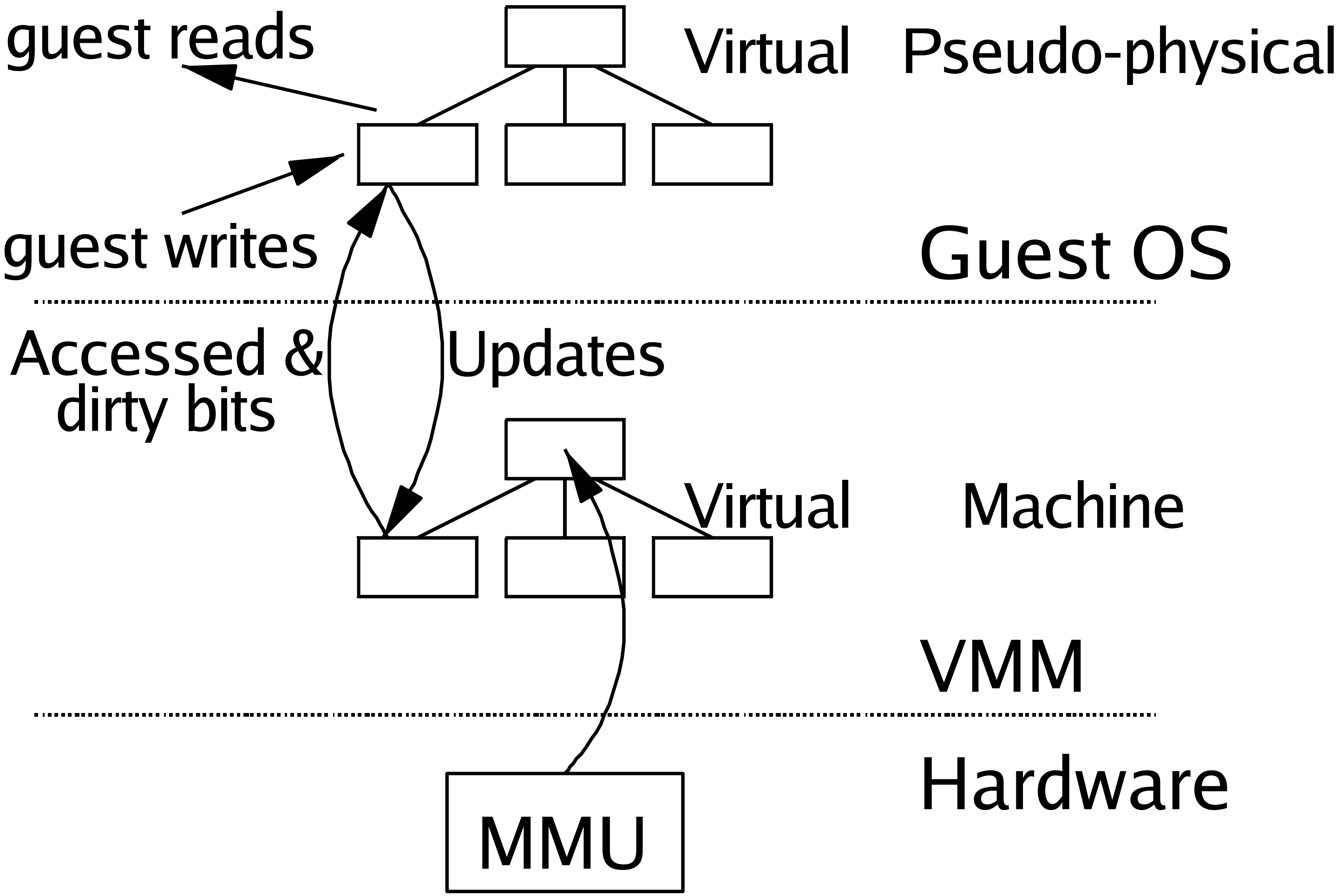
CPU virtualization: x86

- Xen runs in ring 0 (most privileged)
- Ring 1/2 for guest OS, ring 3 for user-space
  - #GP if guest attempts to use privileged instuction
- Xen lives in top 64MB (168MB PAE) of linear address space
  - Andrew has patch queued to allow Linux to make room for Xen
  - Segmentation used to protect Xen as switching page tables too slow on standard x86
- Hypercalls jump to Xen in ring 0
- Linux may install an int80 handler, Xen validates the code segment is ring 1
  - Direct user-space to Linux guest system calls
- Interrupts are handled by Xen, Linux guest uses a lightweight event channel mechanism
- MMU virtualization: shadow vs. direct-mode

MMU Virtualization: x86 Shadow Mode

- Linux guest maintains set of page tables
- Xen hypervisor maintains shadow copy
- Shadow copy is visible to hardware MMU
- Xen propagates changes between guest PT and shadow PT
- Expensive: can double page fault rates and has extra memory overhead
- Simpler for guest: Can view physical memory as contiguous, no need to maintain a mapping between guest pseudo physical memory and machine physical memory, and needed for full virtualization

MMU Virtualization: x86 Shadow Mode

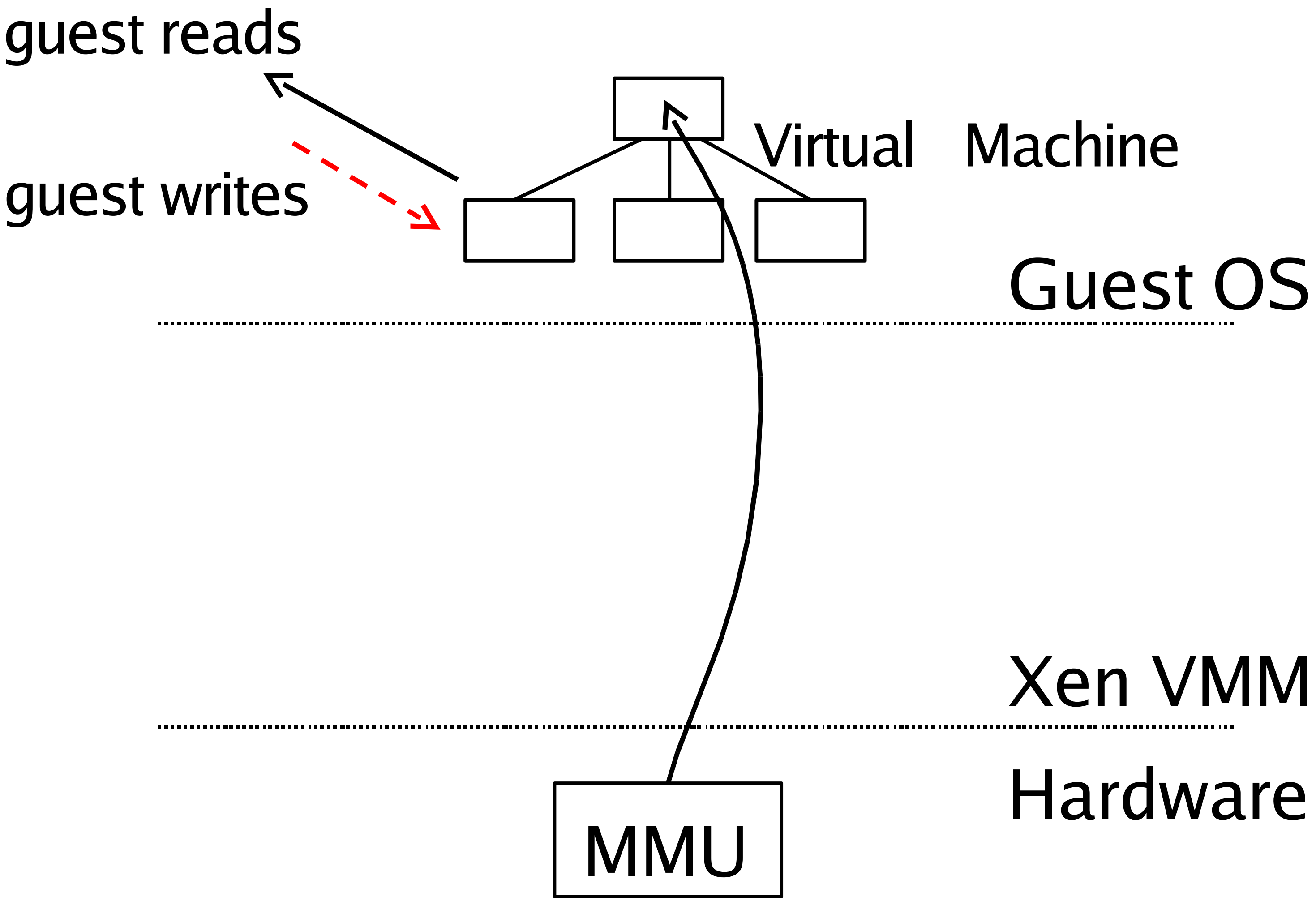


MMU Virtualization: x86 Direct Mode

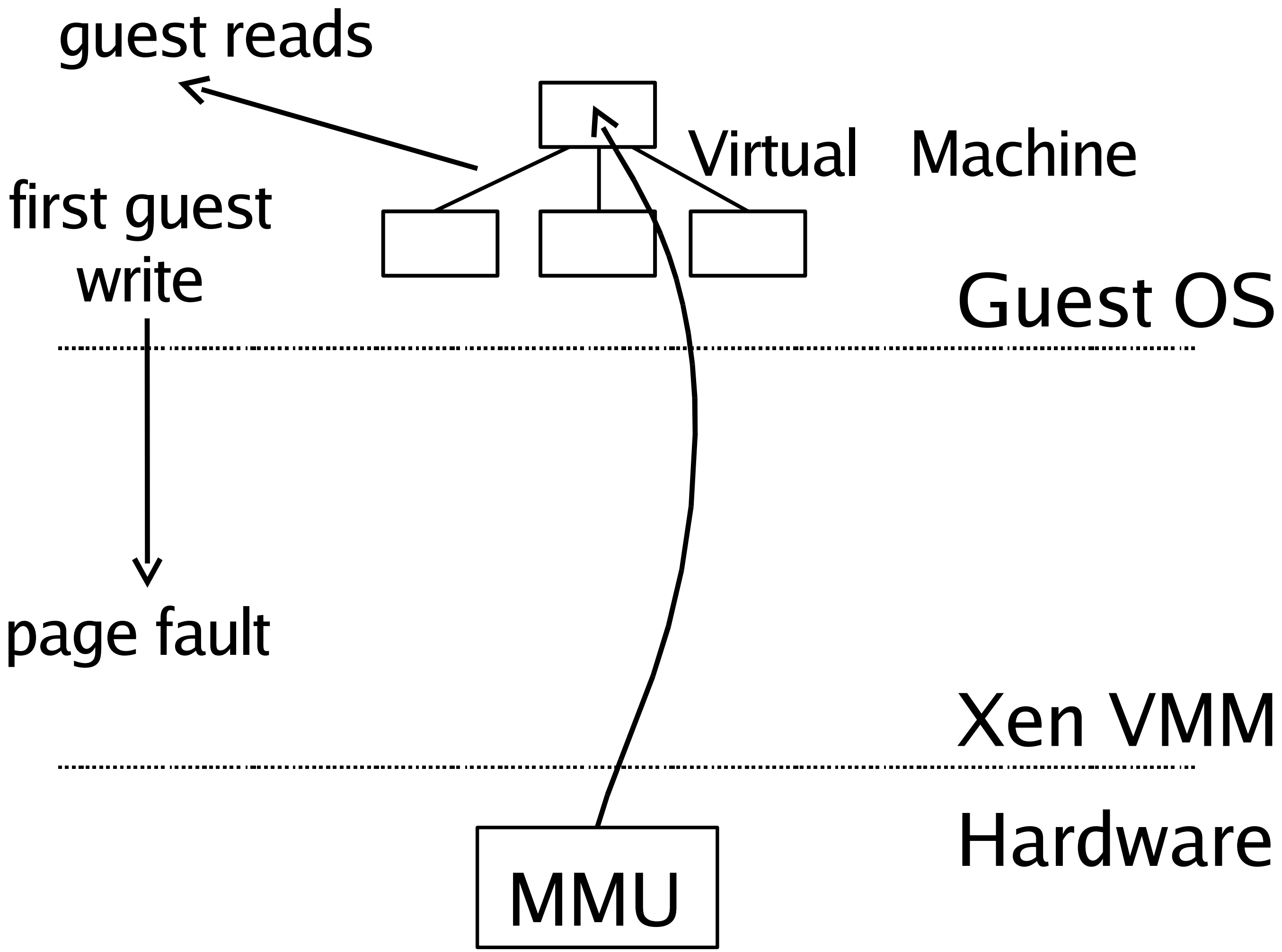
- Linux guest maintains page tables that are visible to MMU
- Linux guest registers pages it will use as page tables with Xen
  - These pages can be one of PD, PT, GDT, LDT, RW (mutually exclusive).
  - Once Xen has pinned a page as a PD or PT it does not need to be revalidated, only updates to it need to be checked (writes will trap).
- Linux uses hypercall to change PT base (e.g. context switch).
- Xen validates page table updates before committing them.
  - Allows incremental updates, avoids revalidation
- Validation rules applied to each PTE:

  1. Guest may only map pages it owns*
  2. Page table pages may only be mapped RO
- Xen traps PTE updates and emulates, or ‘unhooks’ PTE page for bulk updates

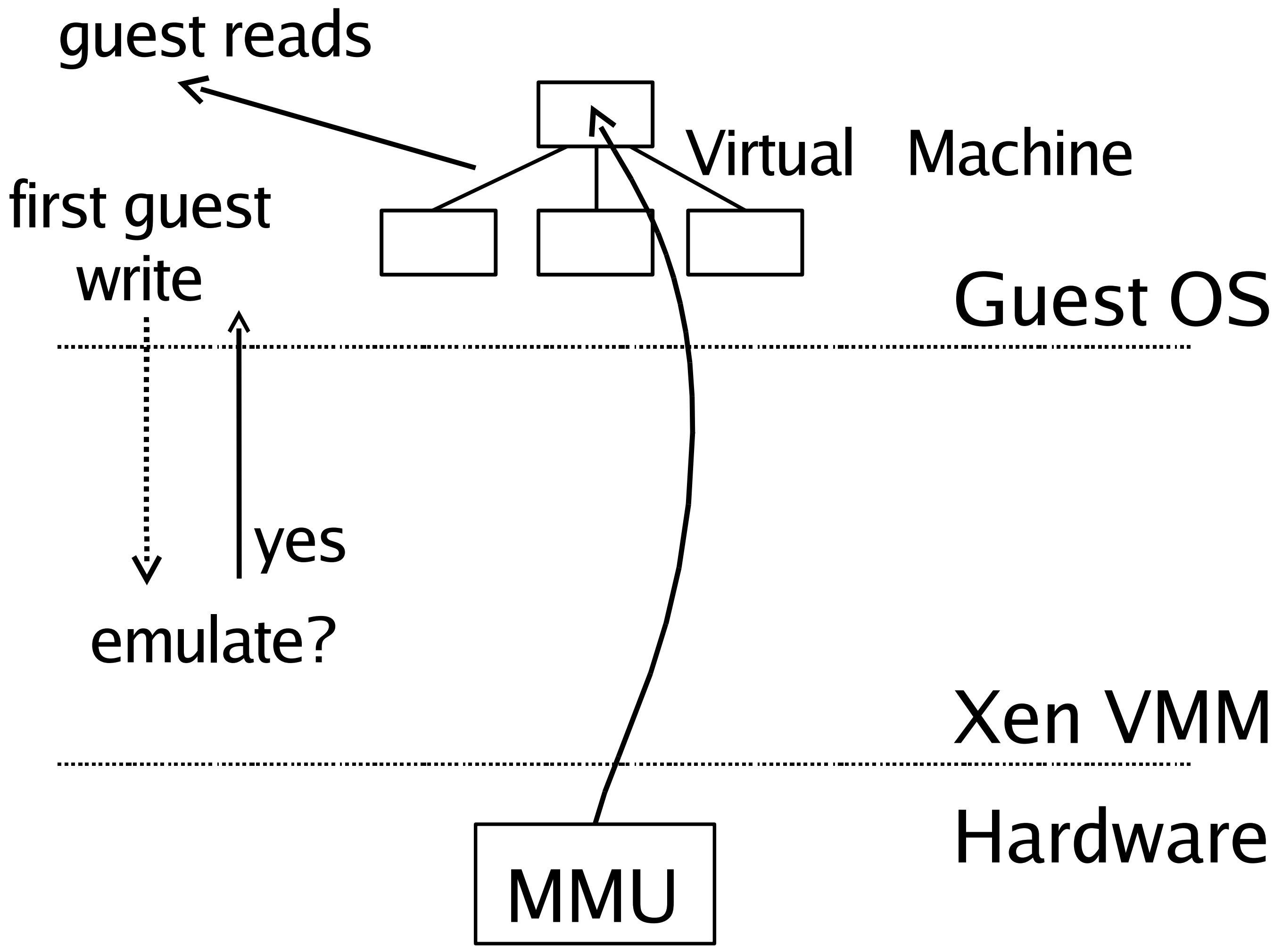
MMU Virtualization: x86 Direct Mode



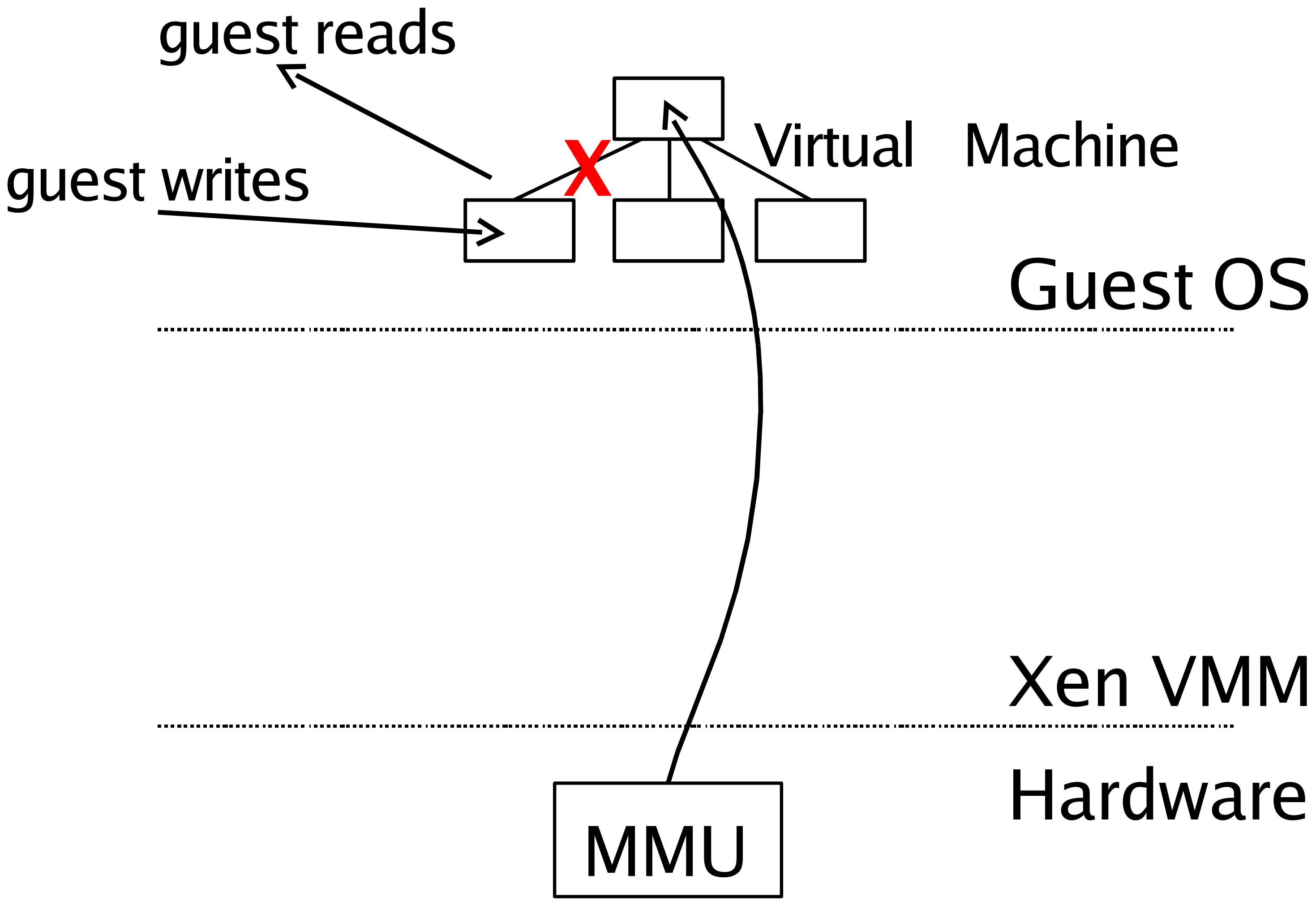
Writable Page Tables: 1 – Write Fault



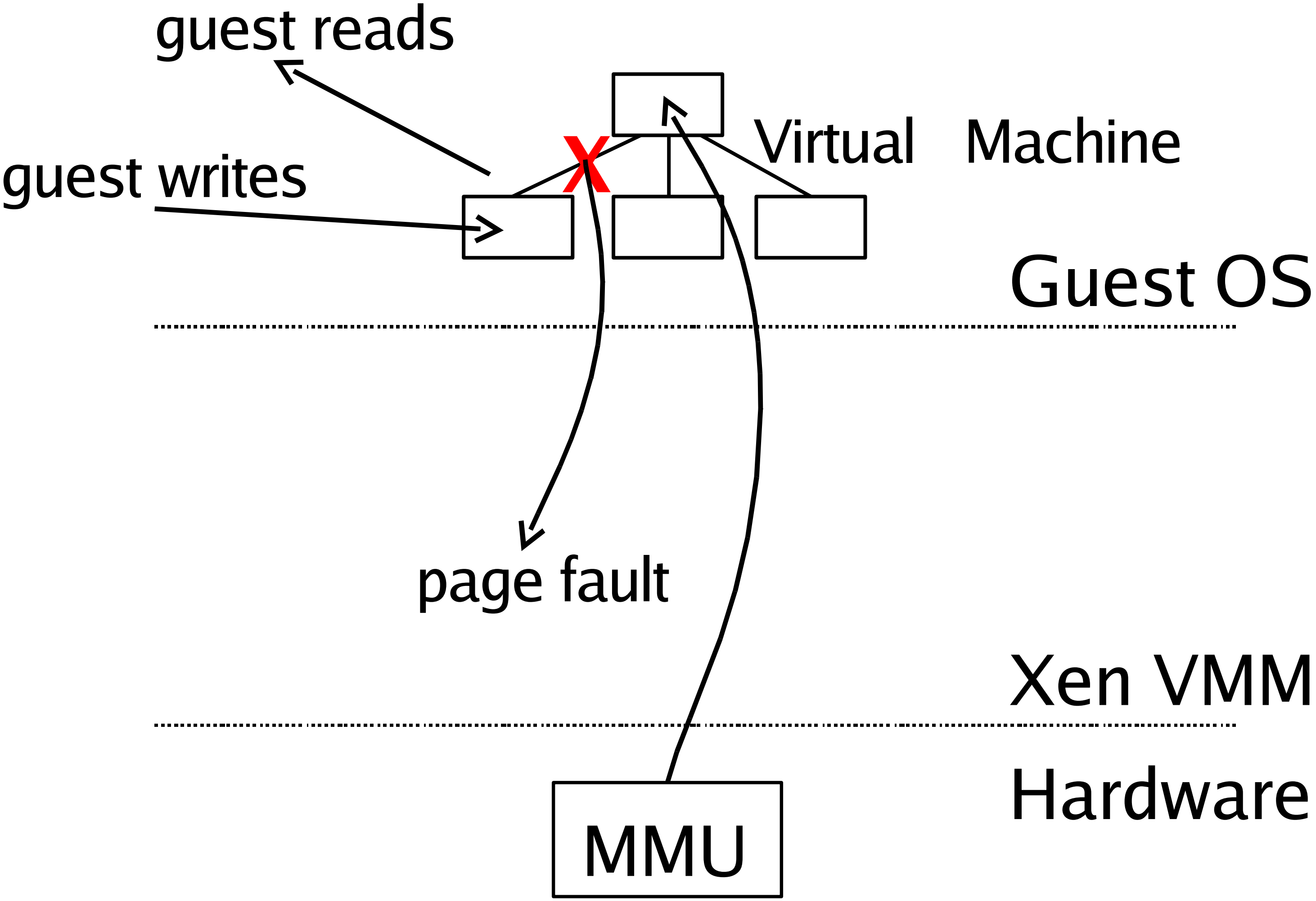
Writable Page Tables: 2 – Emulate?



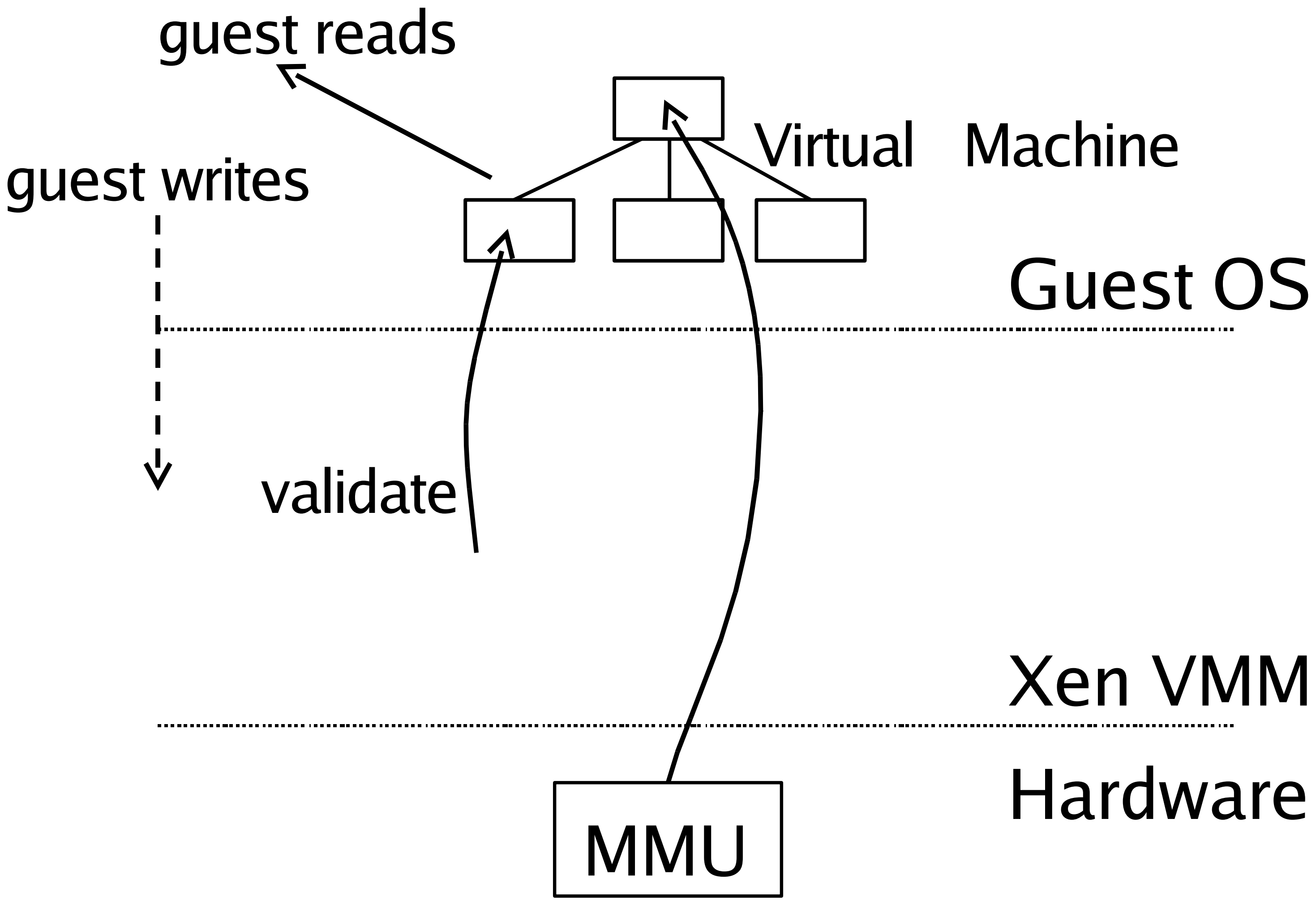
Writable Page Tables: 3 – Unhook



Writable Page Tables: 4 – First Use



Writable Page Tables: 5 – Re-hook



guest reads

guest writes

Virtual Machine

Guest OS

validate

Xen VMM

Hardware

MMU

SMP Guests

- Virtual IPI handled with Xen event channels
  - Important to avoid sending virtual IPI when not necessary
- 32 VCPUs supported on x86
- Simple hotplug/unplug of VCPUs
  - From within VM or via control tools
  - Optimize one active VCPU case by binary patching spinlocks (patch is now in upstream Linux)

I/O Virtualization

Xen *IO-Spaces* delegate guest OSes protected access to specified h/w devices

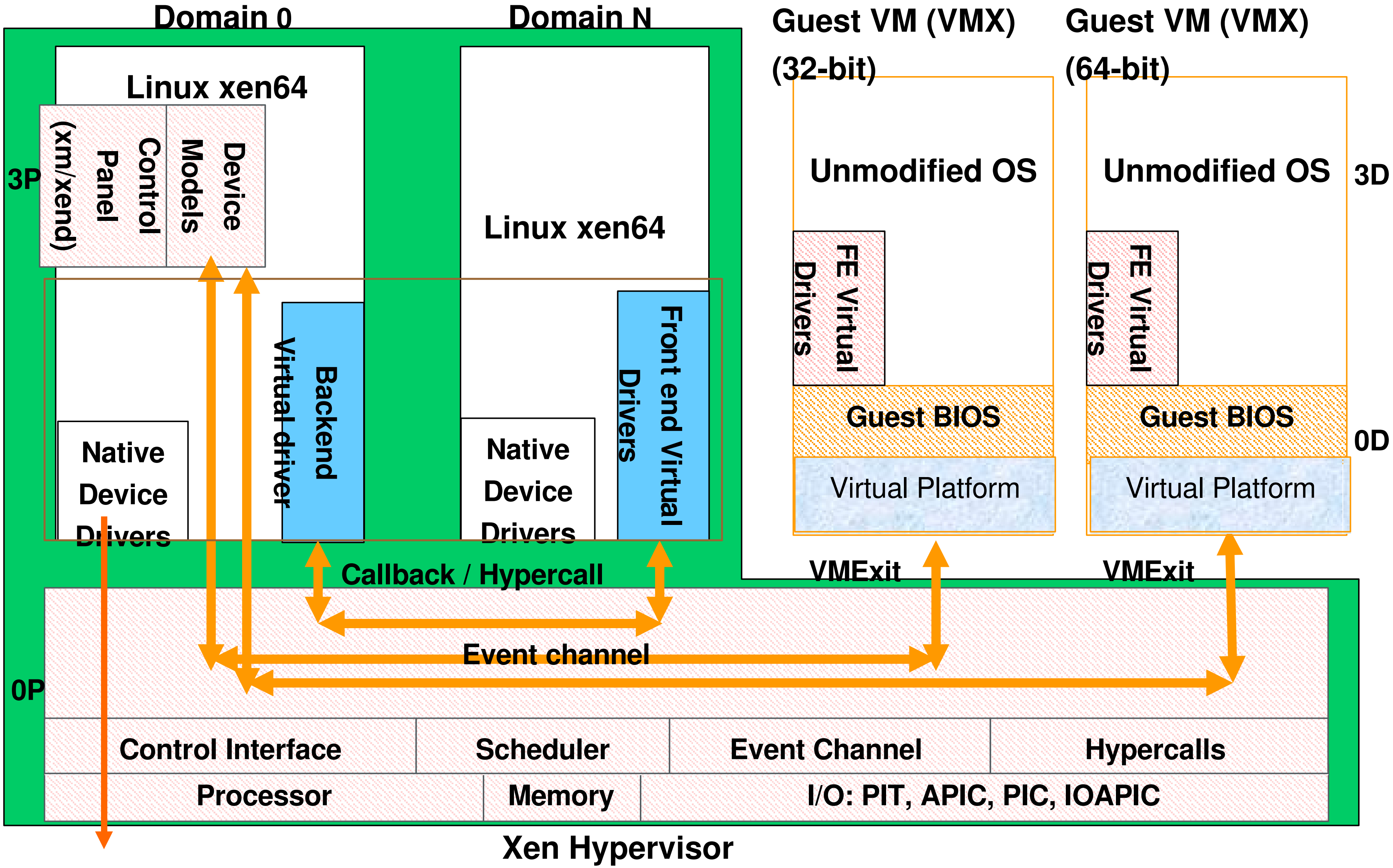
  - Virtual PCI configuration space
  - Virtual interrupts
  - (Need IOMMU for full DMA protection)

- Devices are virtualized and exported to other VMs via *Device Channels*
  - Safe asynchronous shared memory transport built from grant tables and event channels
  - ‘Backend’ drivers export to ‘frontend’ drivers
  - Net: use normal bridging, routing, iptables
  - Block: export any blockk device e.g. sda4,loop0,vg3
- (Infiniband / Smart NICs for direct guest IO)

Full Virtualization: HVM (VT-x, SVM)

- Enable Guest OSes to be run without para-virtualization modifications
  - E.g. legacy Linux, Windows XP/2003
- CPU provides traps for certain privileged instrs
- Shadow page tables used to provide MMU virtualization
- Xen provides simple platform emulation
  - BIOS, Ethernet (ne2k), IDE emulation
- (Install paravirtualized drivers after booting for high-performance IO)

HVM Architecture



Xen Status

- Xen 3.0.0
  - Released January 2006
  - SMP support (SMP hardware and SMP guests)
  - Working ACPI (moved from hypervisor to dom0), Hypervisor time APIs
  - x86_64 (Opteron and EM64T), PAE support (>4 Gb), basic IA64
- Xen 3.0.1
  - Feb 1, 2006
  - Primarily bugfixes and code cleanups
- Xen 3.0.2
  - April 13, 2006
  - HVM now supports VT and SVM
  - 2.6.16 kernel with proper subarch support
- Xen 3.0.x
  - Better driver domains, NUMA support, possible IDC enhancements

XenLinux Merge Status

- Scope of work
  - i386 only
  - UP only
  - domU only
  - shadow mode only
  - Limited scope reduces size, complexity, and invasiveness of the patchset.
- Community response
  - Useful feedback for improving the patchset that has resulted in cleanups which are being propagated back to the xen-unstable developement tree
  - Some small bits have been taken by Andrew for upstream Linux

XenoLinux Merge Status – Patchset details

- ~35 patches, ~1.6MB
- 114 files changed, 13522 insertions(+), 350 deletions(-)
- Creates new i386 subarch: `arch/i386/mach-xen`
- Updates infrastructure to allow a subarch to override default behaviour for:
  - Start-of-day
  - Segments (running in ring 1)
  - Descriptor table handling: GDT, LDT, IDT
  - Control register handling: CR0, CR1, CR2, CR3, CR4
  - CPUID
  - Interrupt handling
  - TLB handling
  - Memory and page table handling
  - Idle loop

XenLinux Merge Status – Patchset details

- Adds core Xen functionality for:
  - Hypervisor interface
  - Time
  - Reboot
  - Event channels
  - Grant tables
  - Xenbus
  - Console
  - Frontend block and net drivers

XenLinux Merge Status – Related Work

- VMI proposal from VMWare
  - Common binary interface layer for hypervisors
  - Pros: Resembles native platform, good native performance, easy to change hypervisors without changing kernels.
  - Cons: Strict ABI, low-level interface may have poorer paravirt performance, no users, requires extra glue layer (the ROM).
- `paravirt_ops` from Rusty Russell
  - Common paravirt function table interface for hypervisors. Similar to VMWare proposal with focus on standard Linux coding practices. Provides an internal kernel API rather than forcing ABI.
  - Pros: Follows common conventions, draws from good aspects of VMI
  - Cons: Early work, still needs to be flushed out, no users

XenLinux Merge Status – Future Work

- Continue to respond to feedback from LKML
- Repost as ready
- Cleaner patch split so that we can easily feed the non-confrontational patches to Andrew. Much of the infrastructure changes are the same for Xen, VMI and `paravirt_ops`.
- Follow-on work
  - SMP support
  - Writable page tables support
  - dom0 support
  - Other architectures (x86_64, ia64, PPC)

Xen Roadmap

- Performance and scalability
  - Fix any performance regressions from Xen 2.0, NUMA support
- IOMMU support
- Get Xen upstream ;-)
- Improved resource control
  - Fine grained delegations, dynamic VCPU to CPU binding
- Network drivers support for S/G and TSO/UFO
- HVM improvements
  - Shadow page table improvements
  - QEMU: VNC Server, USB Mouse, Virtual Framebuffer
  - SMP HVM guests
  - New I/O model for HVM guests
- And much, much more. Come join in the fun!