Xen and the Art of Virtualization

Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt & Andrew Warfield

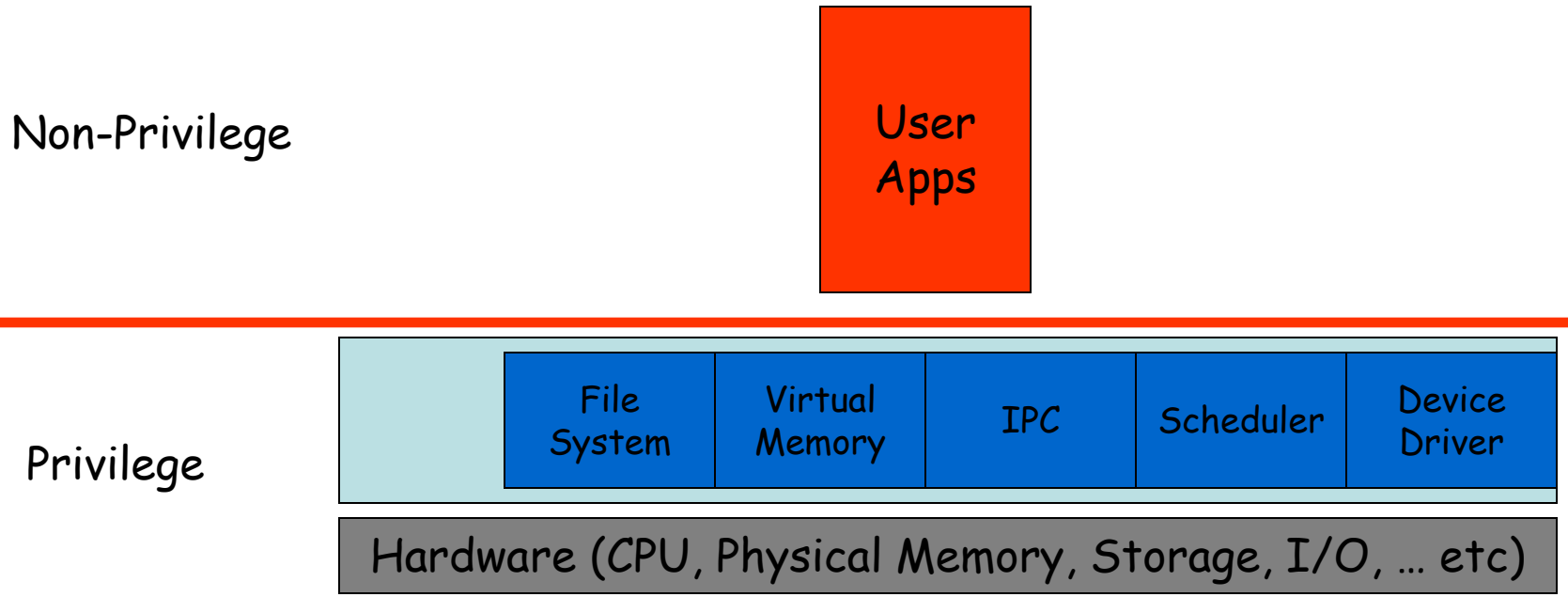
Presented by Anthony So
November, 13 2013

Presentation Overview

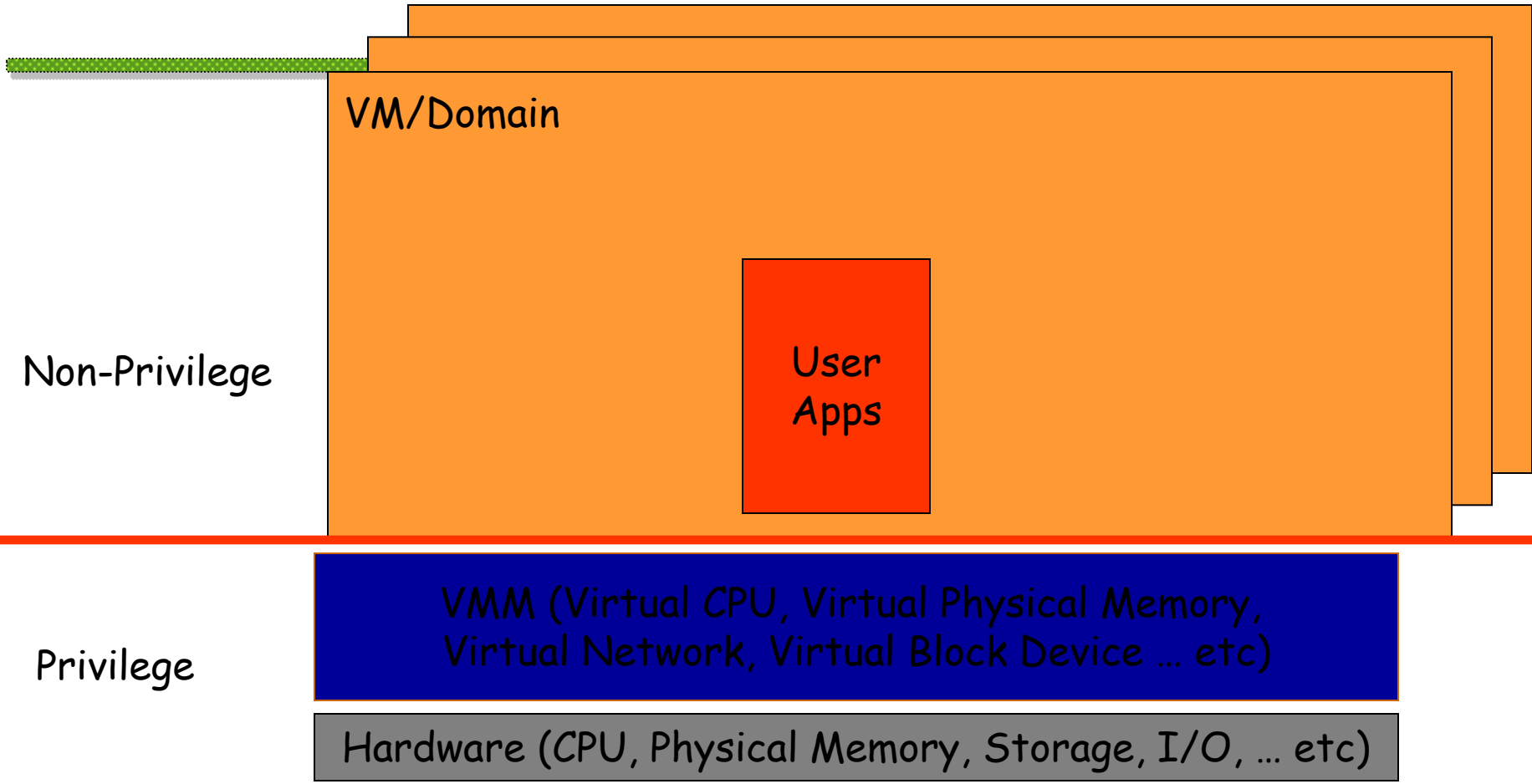
- Introduction
- Xen approach
  - Overview
  - Implementation
  - Evaluation
- Summary

Introduction

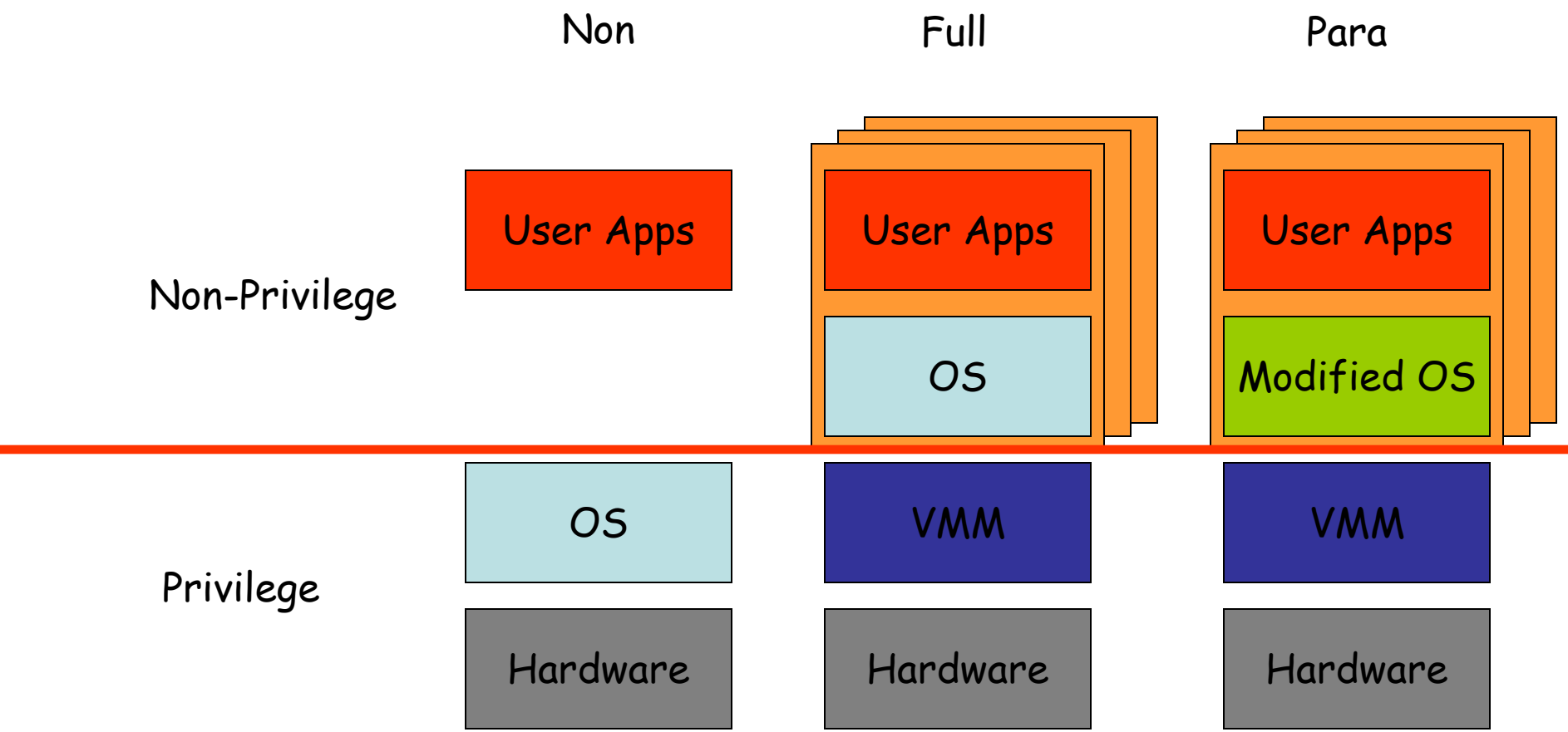
Monolithic kernel



Virtualization

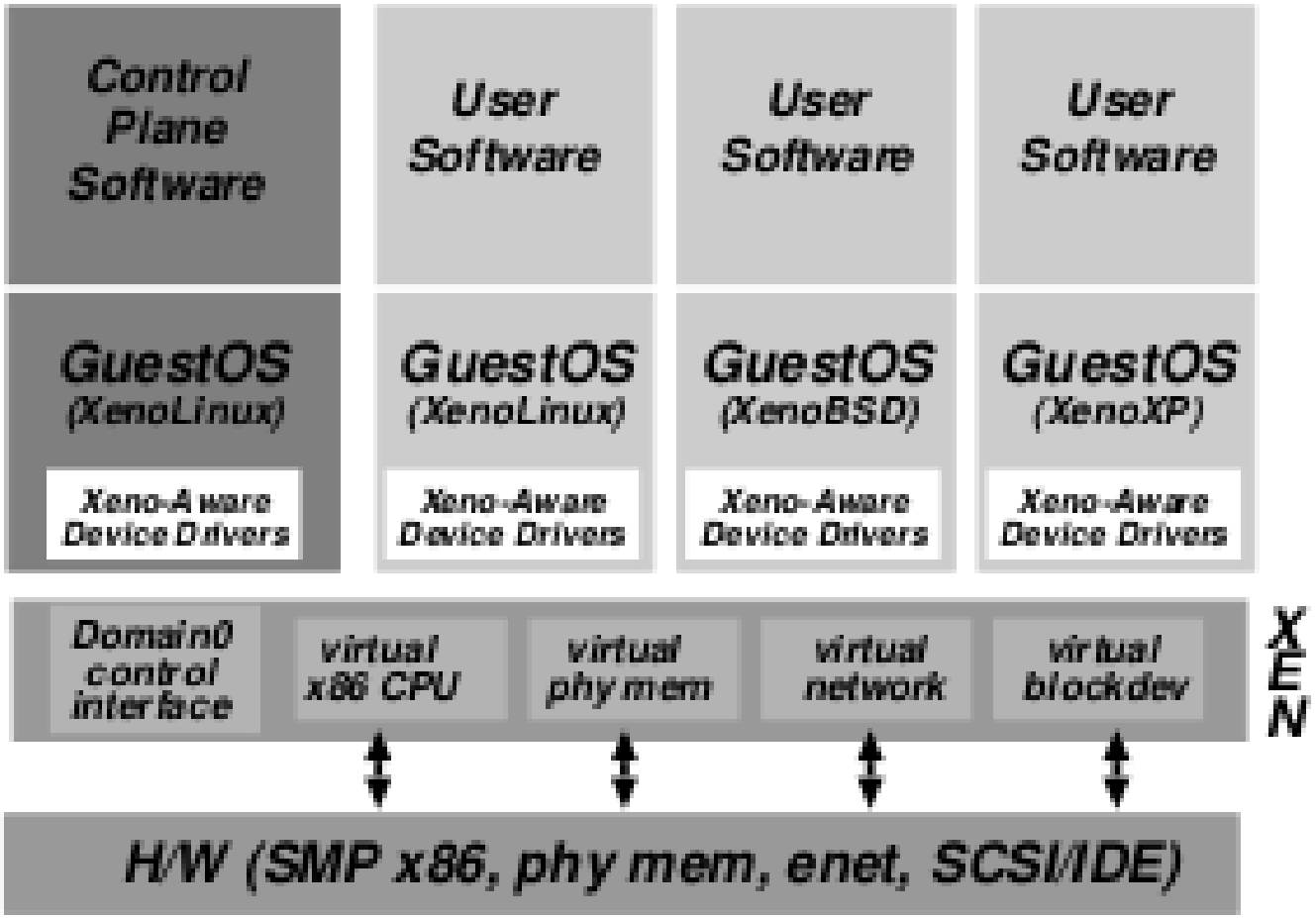


Non, Full, and Para-Virtualization



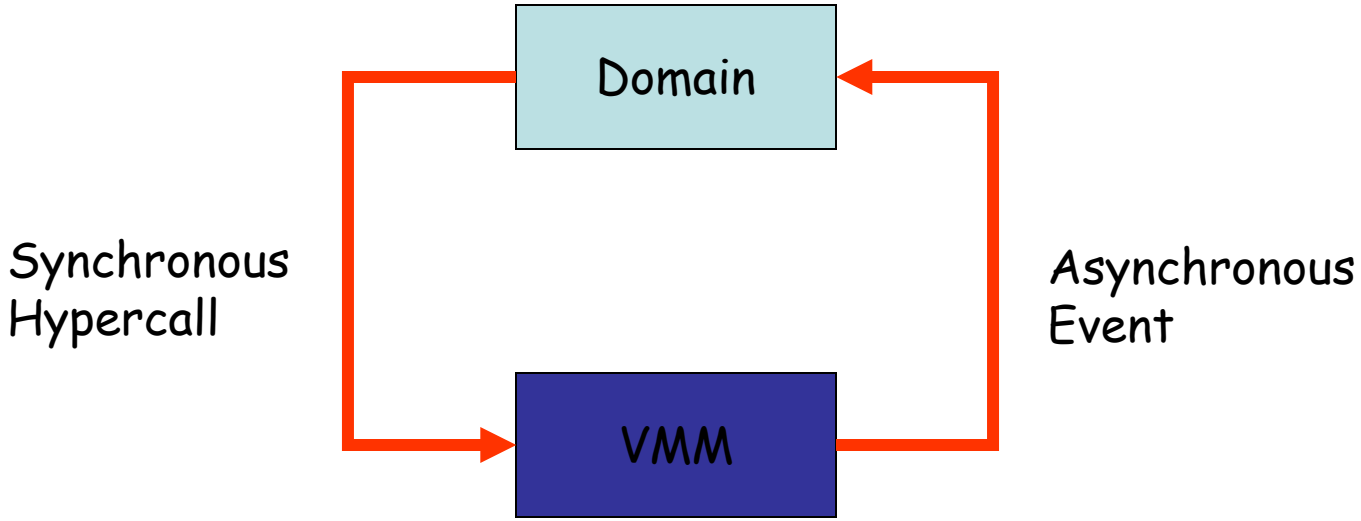
Xen - Overview

Xen Architecture Overview



Control Transfer

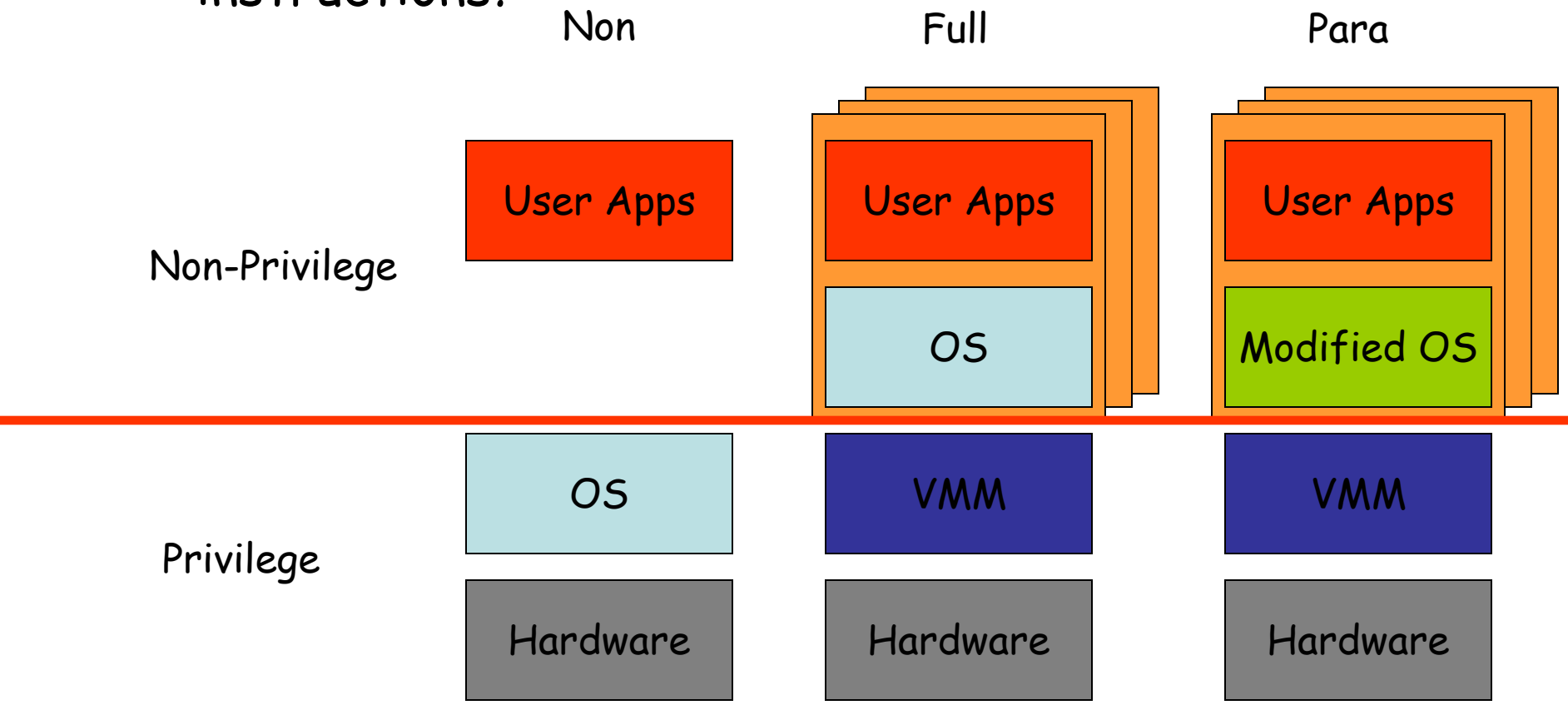
- Synchronous calls from a domain to Xen may be made using a hypercall
- Notification are delivered to domains from Xen using an asynchronous event mechanism



Xen – Implementation

CPU – Privilege Instruction

- How x86 architecture handles privileged instructions?



Memory Management

- Tagged TLB vs No Tagged TLB
- Tagged TLB is ideal for virtualization because each TLB entry associated with an address-space identifier to allows hypervisor and guest OS entries to coexist even with context switch, thus, avoid complete TLB flush.
- x86 – No Tagged TLB and must flush after a context switch.

- Xen exists in a 64MB section a the top of every address space, thus avoiding a TLB flush when entering and leaving the hypervisor.

Memory Management

- S/W managed vs H/W managed TLB

- x86 uses H/W managed TLB. Therefore, TLB management and handling TLB faults are done entirely by the MMU hardware.

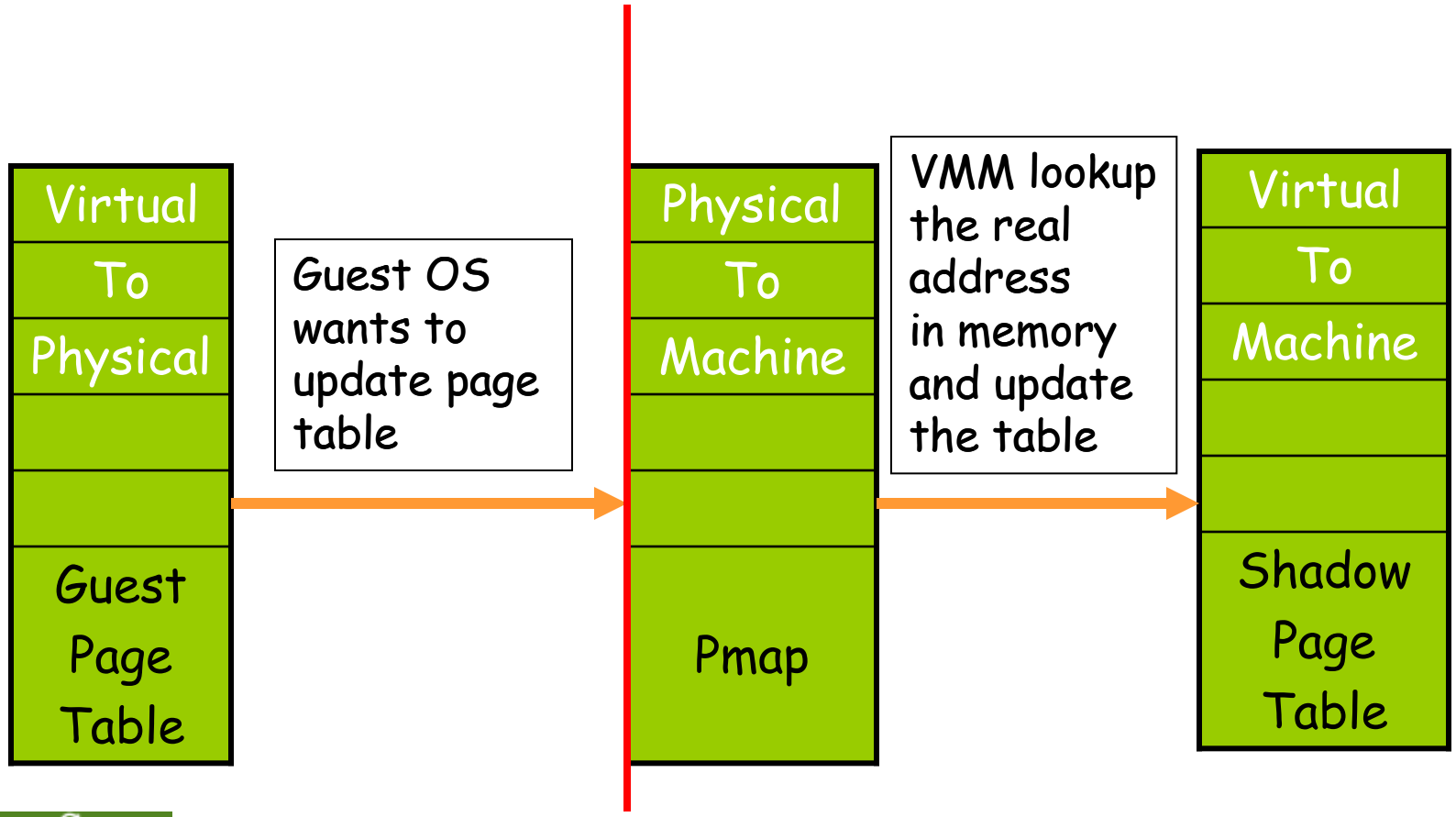
- S/W managed TLB is ideal for virtualization because TLB misses are serviced by the OS.

Memory Management

- Xen register guest OS page tables directly with the MMU but restricted guest OS to read-only access.
- Page Table updates are passed to Xen via hypercall.
- Request are validated before being applied.
  - Type: writable, page table … etc.
  - Reference count: Must be 0 to switch task type.
- To minimize hypercall, guest OS locally queue updates before applying an entire batch with a single hypercall.

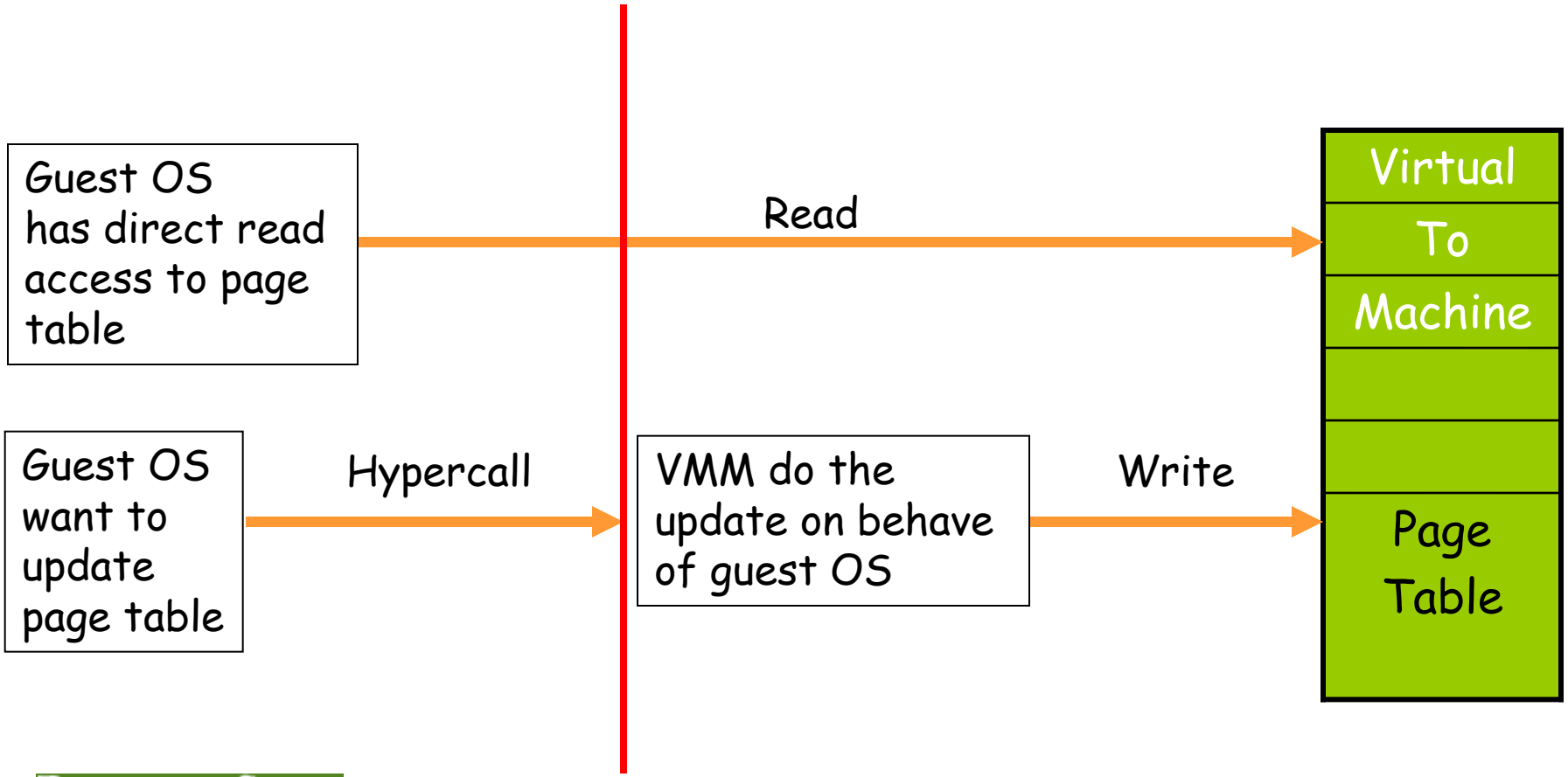
Memory Management

- Shadow Page Table.



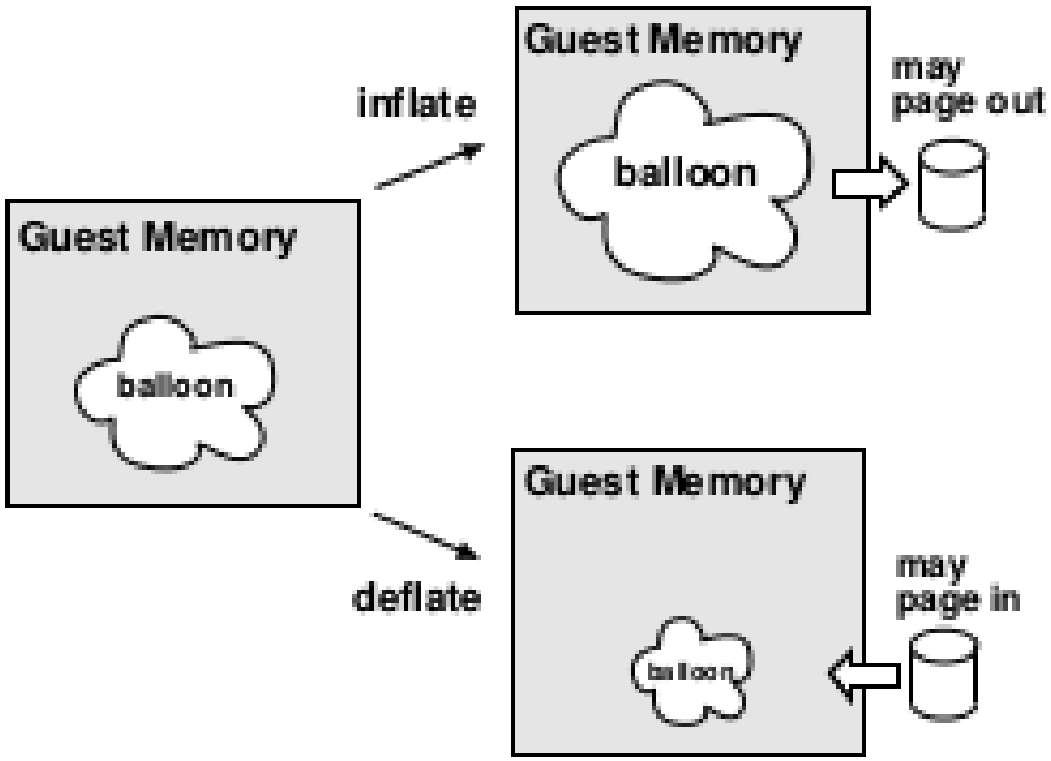
Memory Management

- Xen



Memory Management [3]

- Balloon Driver is a mechanism to adjust a domain's memory usage.



Exception / System Calls / Interrupt

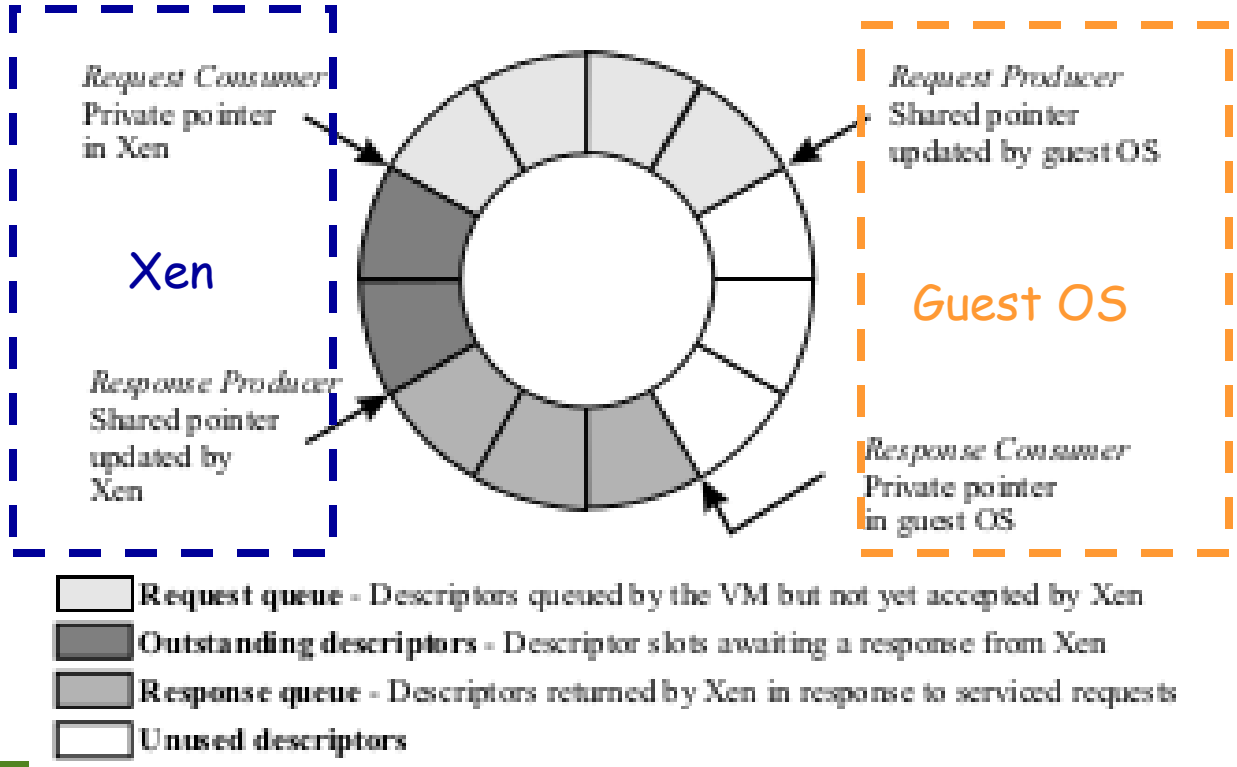
- Exception: A table describing the handler for each type of exception is registered with Xen for validation. The handler are identical to real x86 hardware (except page faults).
- System Calls: Xen allows each guest OS to register & install a fast handler to enable direct calls from user apps into its guest OS and avoid routing through Xen on every calls.
- Interrupt: Hardware interrupts are replaced with a lightweight event system.

Time and Timers

- Xen provides guest OS the following notion of time:
- Real Time:
  - Time that is maintained continuously since machine boot.
- Virtual Time:
  - Time that a particular domain has executed. It will not advance if the domain is not executing.
- Wall-Clock Time:
  - Current Real Time + an offset.

I/O Ring

- An asynchronous I/O rings is used for data transfer between Xen and guest OS. (Circular queue)



Network

- Xen provides the following abstraction:
- Virtual firewall-router (VFR)
- Virtual network interfaces (VIF) – Like a modem network interface card
- Two I/O rings: transmit and receive.
- Round-Robin packet scheduler.
- Page flipping: require guest OS to exchange an unused page frame for each packet it receives to avoid copying between Xen and the guest OS (but require page-alignment).

Disk

- Domain0 has unchecked access to physical disks.
- All other domains access persistent storage through Virtual block device (VBD).
- Domain0 manages VBDs.
- Ownership and access control information are accessed via the I/O ring.
- Round-round scheduler.
- Batching of requests for better access performance.

Xen - Evaluation

Hardware

- Dell 2650 dual processor 2.4GHz Xeon server
- 2GB RAM
- Broadcom Tigon 3 Gigabit Ethernet NIC
- Hitachi DK32EJ 146GB 10k RPM SCSI disk
- Linux version 2.4.21
- RedHat 7.2

Virtualization Comparison

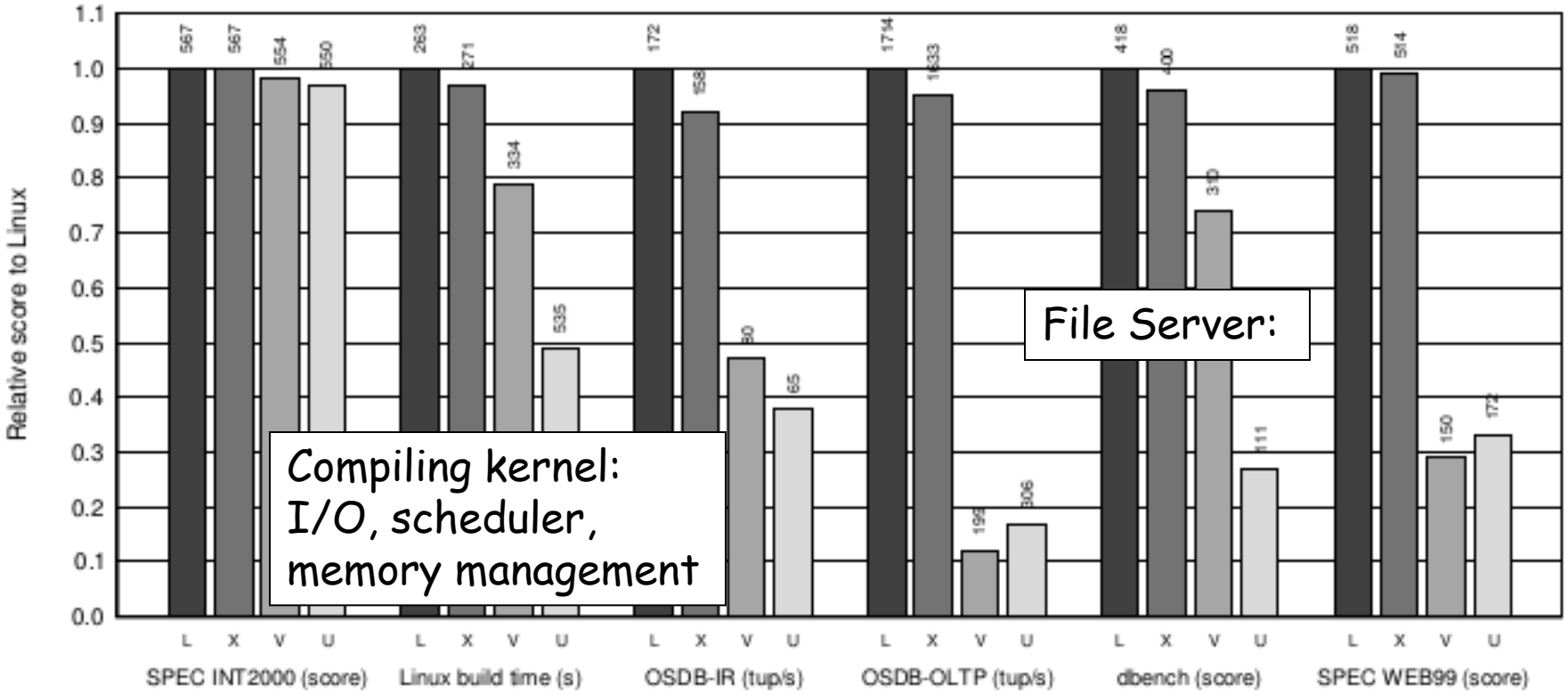
- Native Linux
  - Compiled for i686
- XenoLinux
  - Compiled for Xeno-i686 for Xen
- VMware Workstation
  - Compiled for i686
- User-mode Linux (UML)
  - Compiled for um for UML

Relative Performance

Computation Intensive:
Processor & memory
w/ minimal I/O or O/S

Database:
Sync. Disk operation

Web server:



File Server:

Compiling kernel:
I/O, scheduler,
memory management

Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

Concurrent

- Higher overhead from single domain is due to lack of support to SMP guest OS

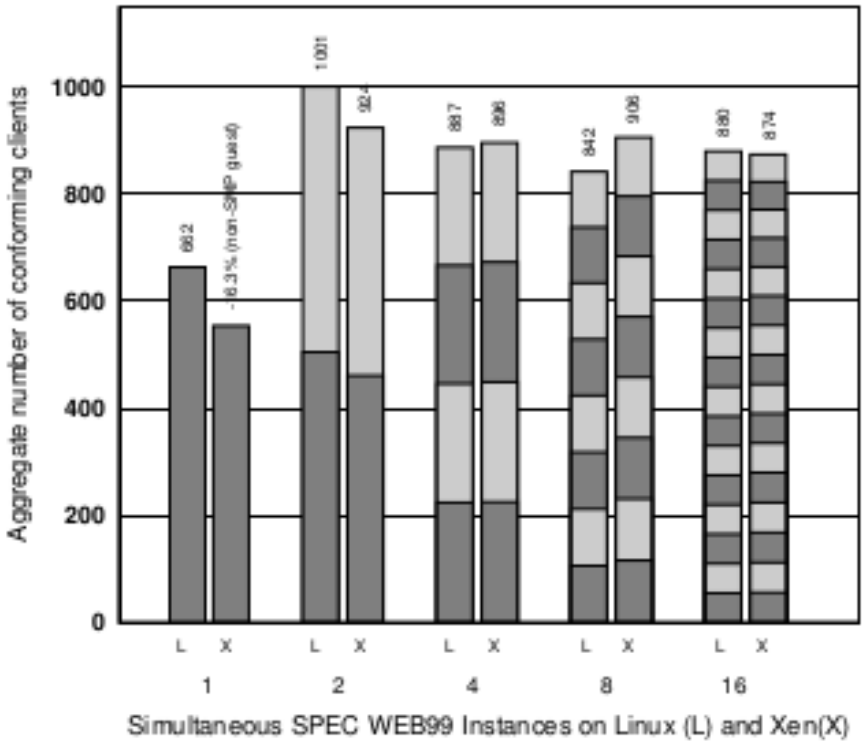


Figure 4: SPEC WEB99 for 1, 2, 4, 8 and 16 concurrent Apache servers: higher values are better.

Conclusion

- Xen is a paravirtualization
- Xen exposes an hypercall interface to Guest OS. Guest OS use it to communicate with Xen to do privileged instructions.
- As a result, Xen can not use unmodified guest OS.
- Performance is comparable to native Linux.

Learn More

- The Xen Project at www.xenproject.org