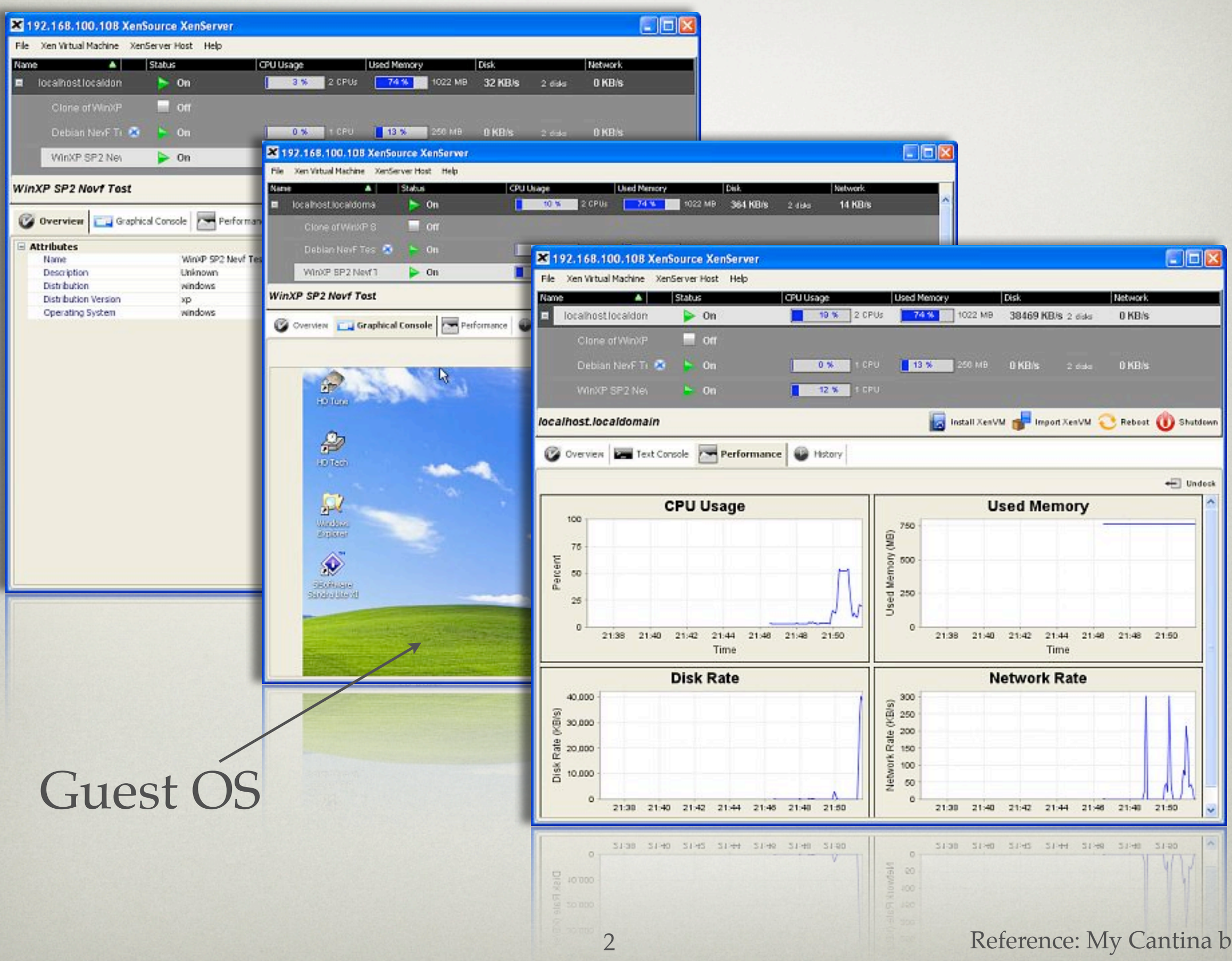
Xen and the Art of Virtualization

Paul Barham et al.
University of Cambridge
Computer Laboratory

Presenter: Hao



Guest OS

Reference: My Cantina blog

Goals

- Virtual Machine should be isolated
  - Adversely affecting another
- Support different types of OS's to accommodate the heterogeneity of different applications
- Small Overhead

Xen

- Secure isolation between VMs
- Resource control and QoS
- Only guest kernel needs to be ported
  - User-level apps and libraries run unmodified
  - XP, Linux 2.4/2.6, *NetBSD, FreeBSD, Plan9, Solaris*
- Execution performance close to native
- *Live Migration of VMs between Xen nodes*
- *Xen hardware support:*
  - *SMP; x86 / x86_64 ; all Linux drivers*

* Xen 3.0

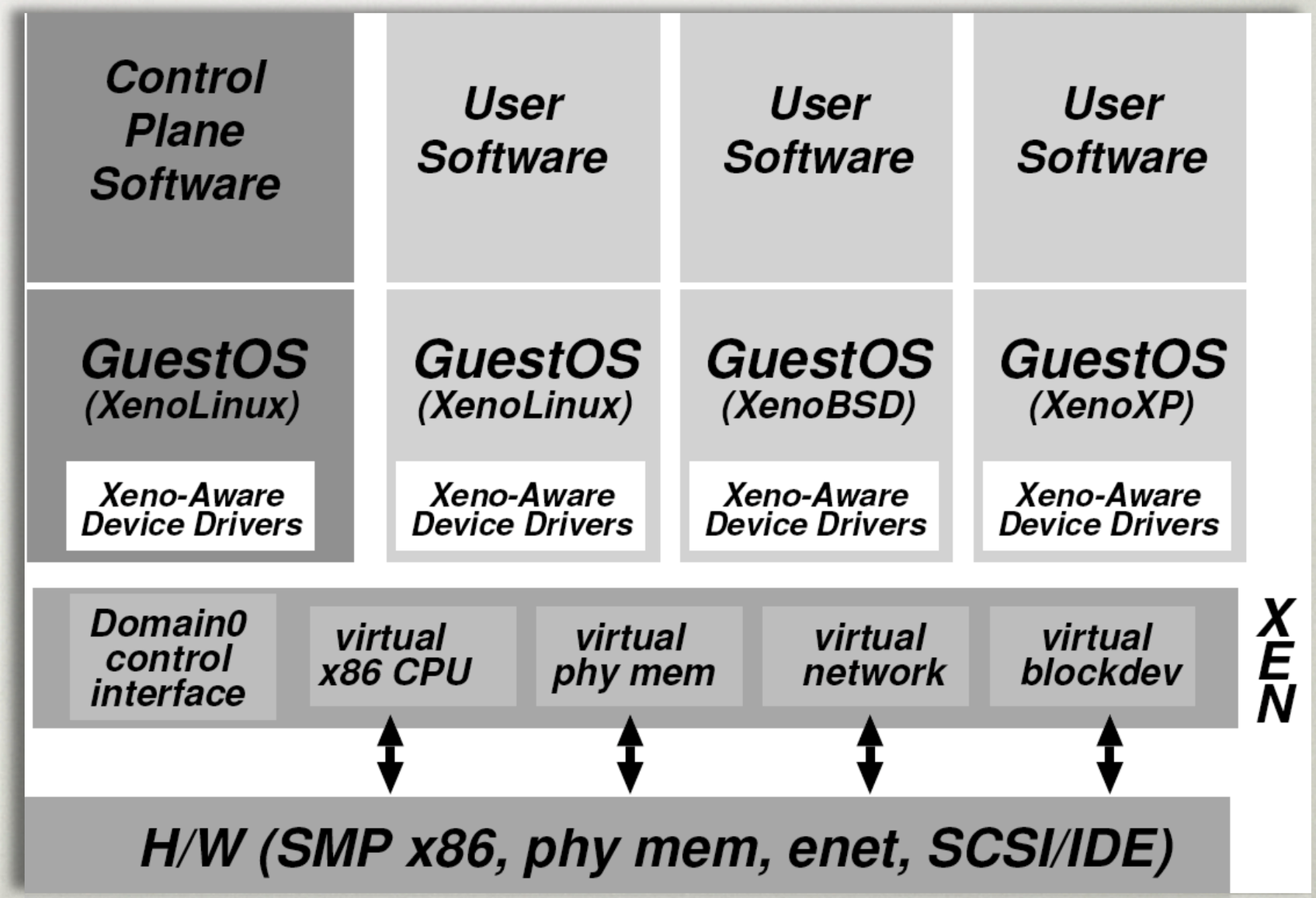
Xen 1.2 Architecture

Unmodified User-Level Application Software

Ported "Guest" OS's

Xen Hypervisor

Hardware



Reference: Barham et al. 2003

Paravirtualization

- Full virtualization: Virtual resources
- Paravirtualization: Real & Virtual
  - Enables OS to optimise behaviour
- TCP timeouts & RTT estimates
- Real machine address: allows guest OS to improve performance

Xen vs. VMware

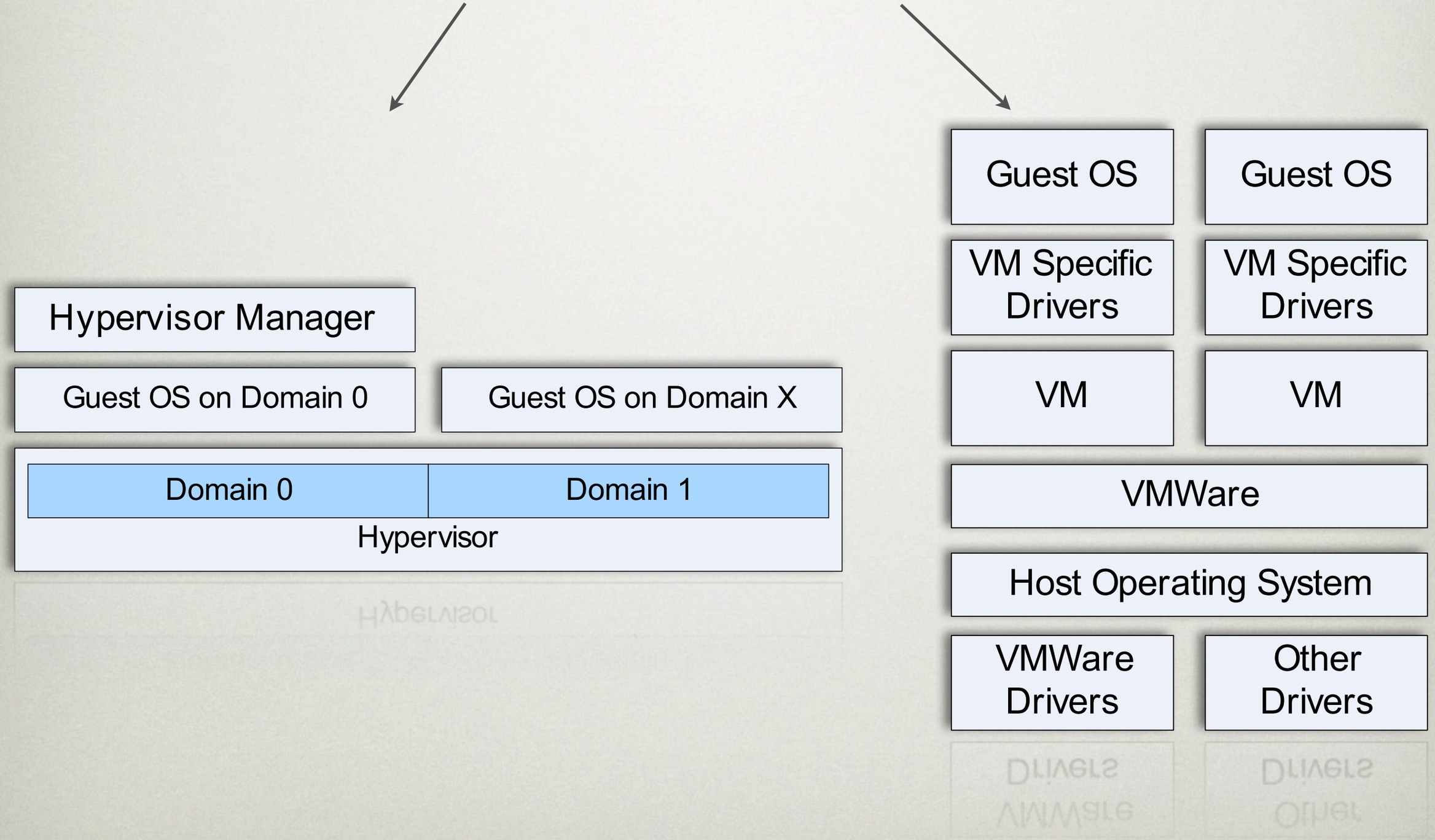


Diagram Reference: Uhlig 04

VMware vs. Xen

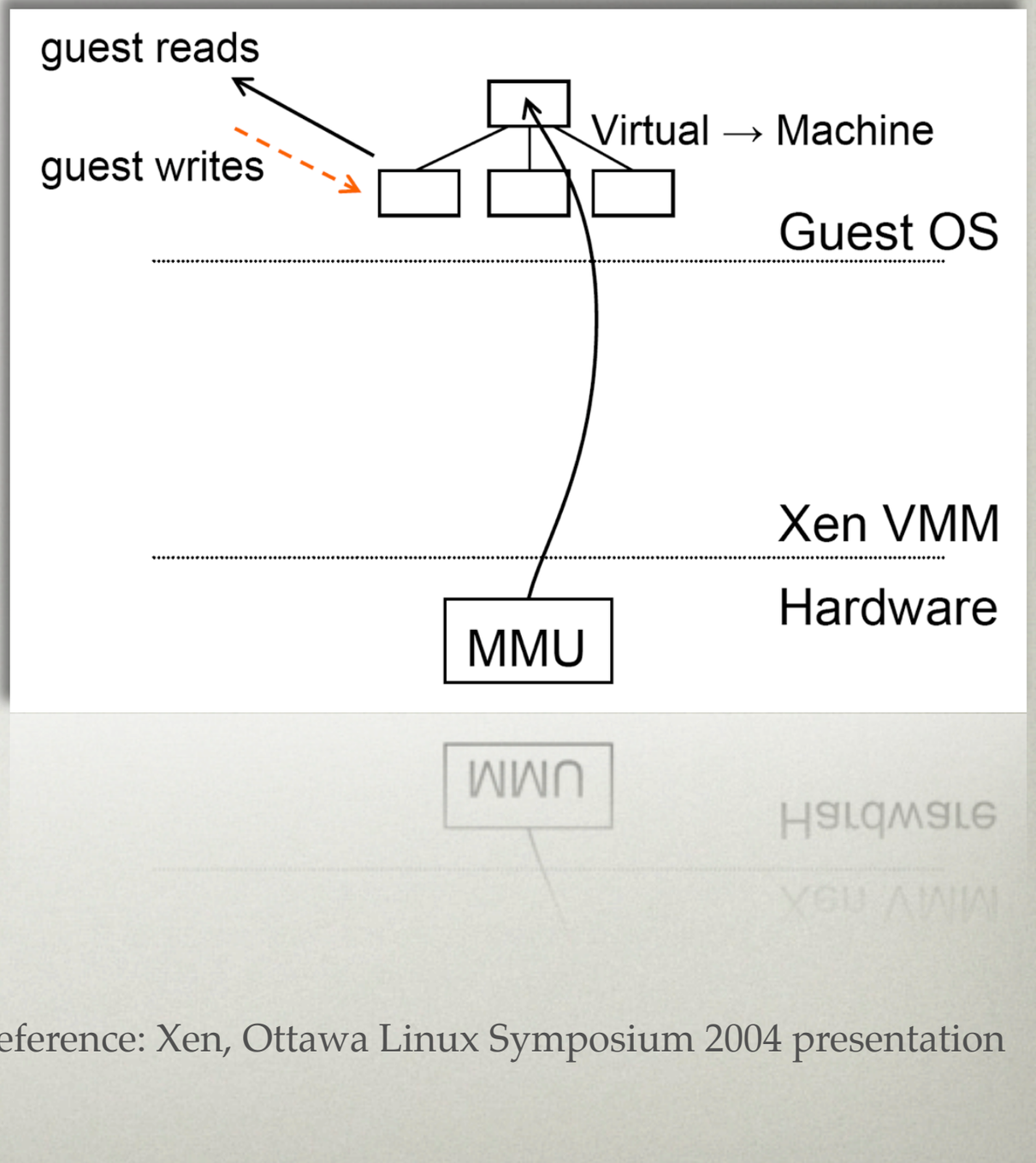
- VMware: OS is not modified, Xen is.
- VMware portable, reliable, safe and easy
- Maturity:
  - VMware has been delivering ESX hypervisor since 2000, Xen just started.

VMware vs. Xen

- Xen is paravirtualization
- Xen is faster - percievable?
- unproven Xen hypervisor
  - Very basic management interface
  - Create, delete, modify VM
  - No performance monitoring
  - No rights management
  - No live migration

Para-Virtualizing the MMU

- Guest OSes allocate and manage own PTs
- Xen must validate PT updates before use
- Validation rules applied to each PTE
- Xen tracks page ownership and current use



Reference: Xen, Ottawa Linux Symposium 2004 presentation

x86 Architecture

- most commercially successful CPU architecture
- Countless computer software
- MS-DOS and Microsoft Windows to Linux, BSD, Solaris OS, and Mac OS X
- Xen

x86 CPU virtualization

- Easier, why? ->
- Has built in security levels (Rings 0, 1, 2, 3)
- Ring 0 – OS Software (most privileged)
- Ring 3 – User software
- Ring 1 & 2 – Not used

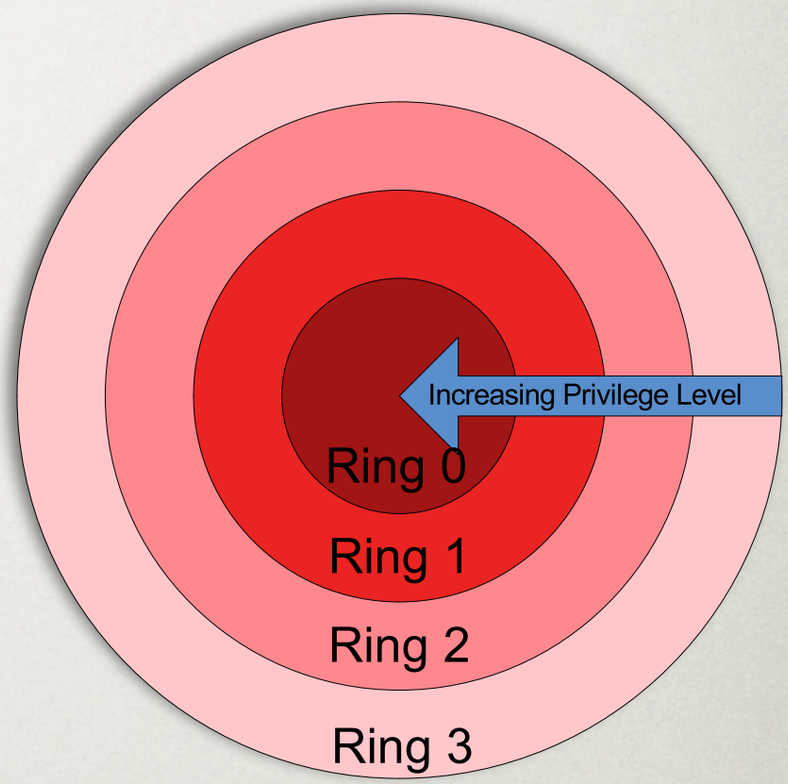
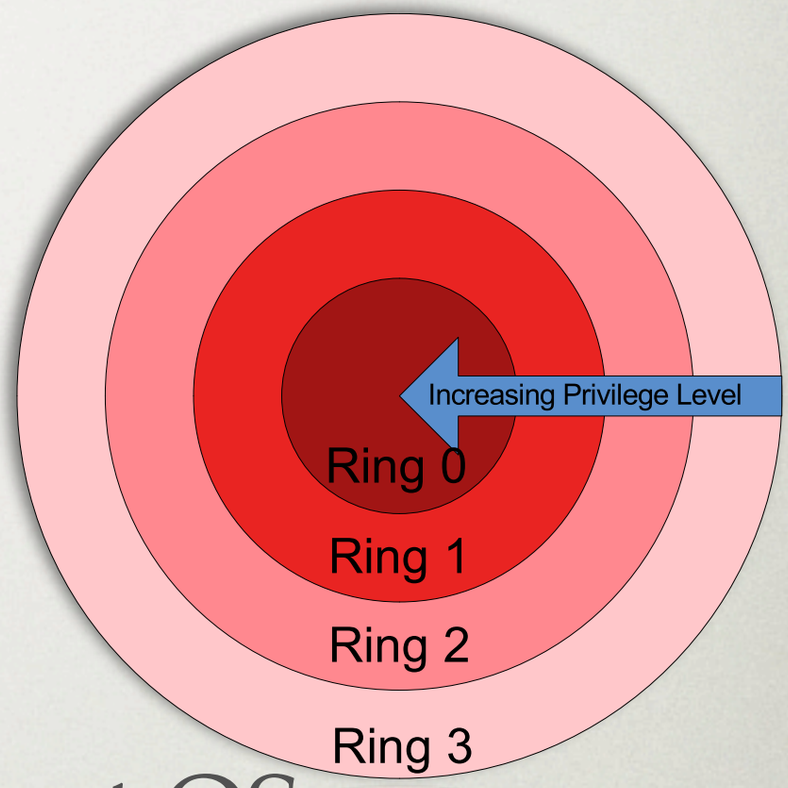


Diagram Reference: Uhlig 04

CPU virtualization

- Xen runs in ring 0 (most privileged)
- Hypercalls jump to Xen in ring 0
- Most changes done on guest OS
- Instructions are paravirtualized by requiring to be validated/executed within Xen



I/O Paravirtualization

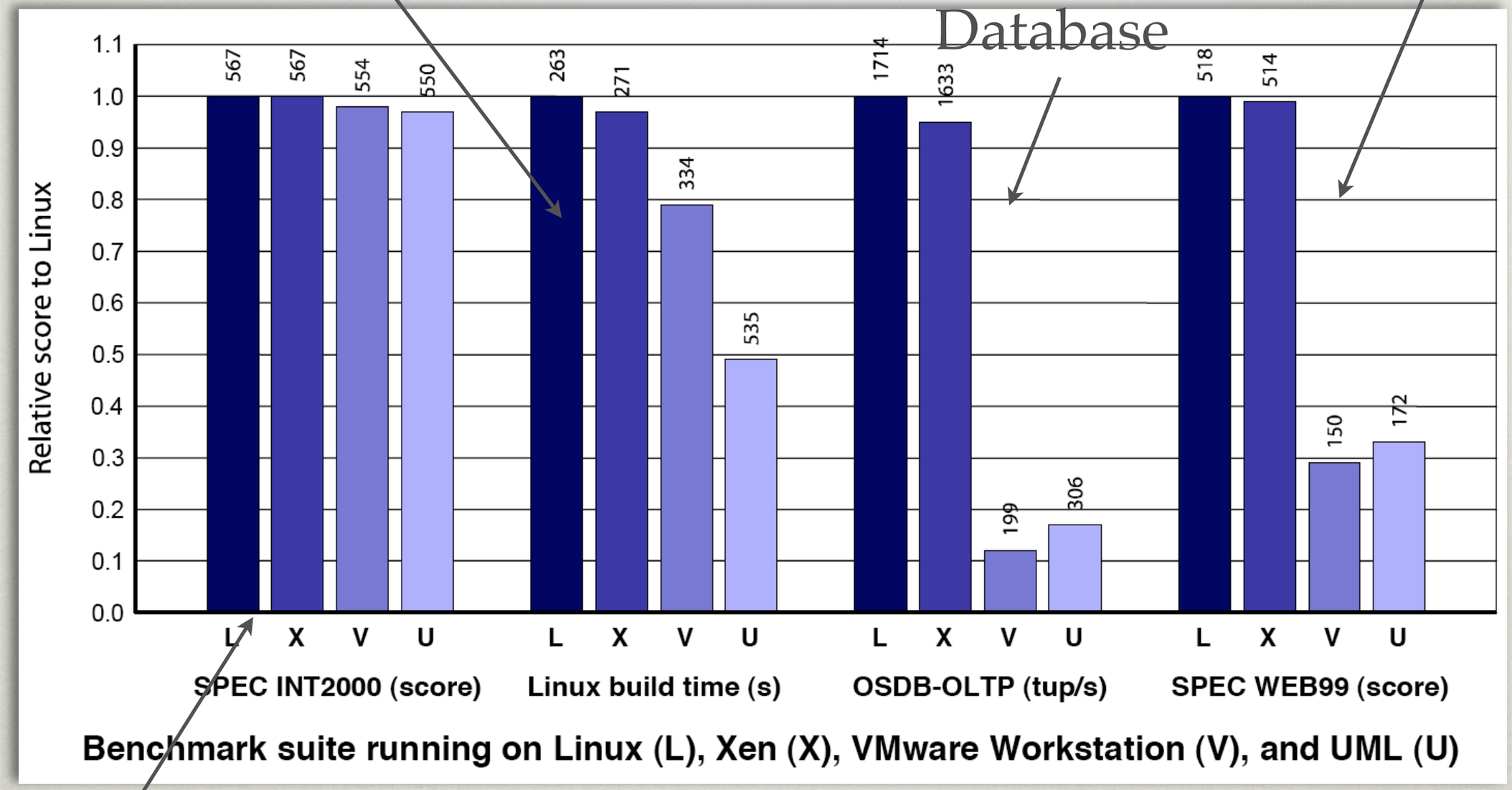
- Xen IO-Spaces delegate guest OSes protected access to specified h/w devices
- Devices are virtualized and exported to other VMs via Device Channels
    - Safe asynchronous shared memory transport

Performance

file I/O, scheduling , MM

Web server, file system, network

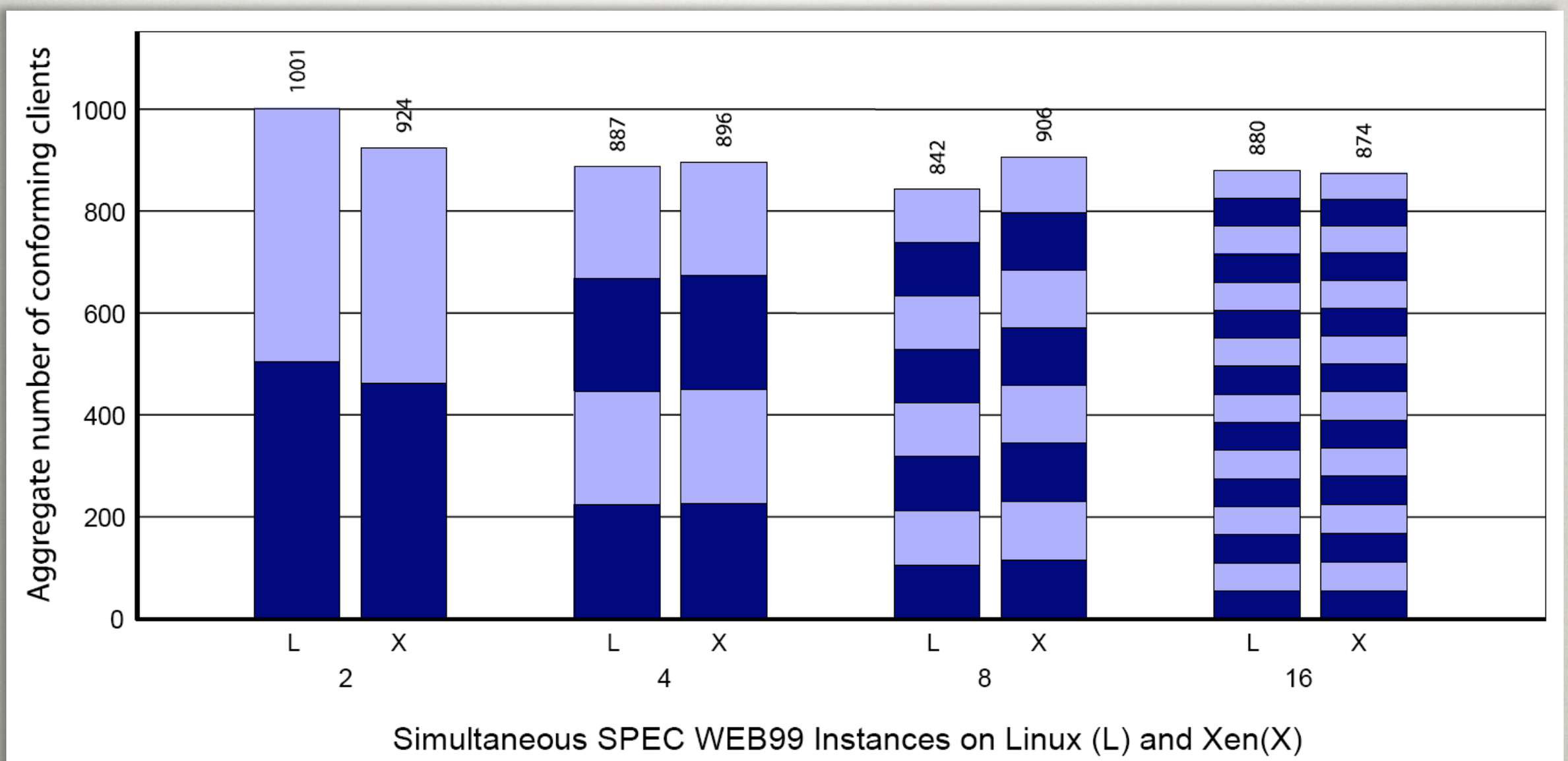
Database



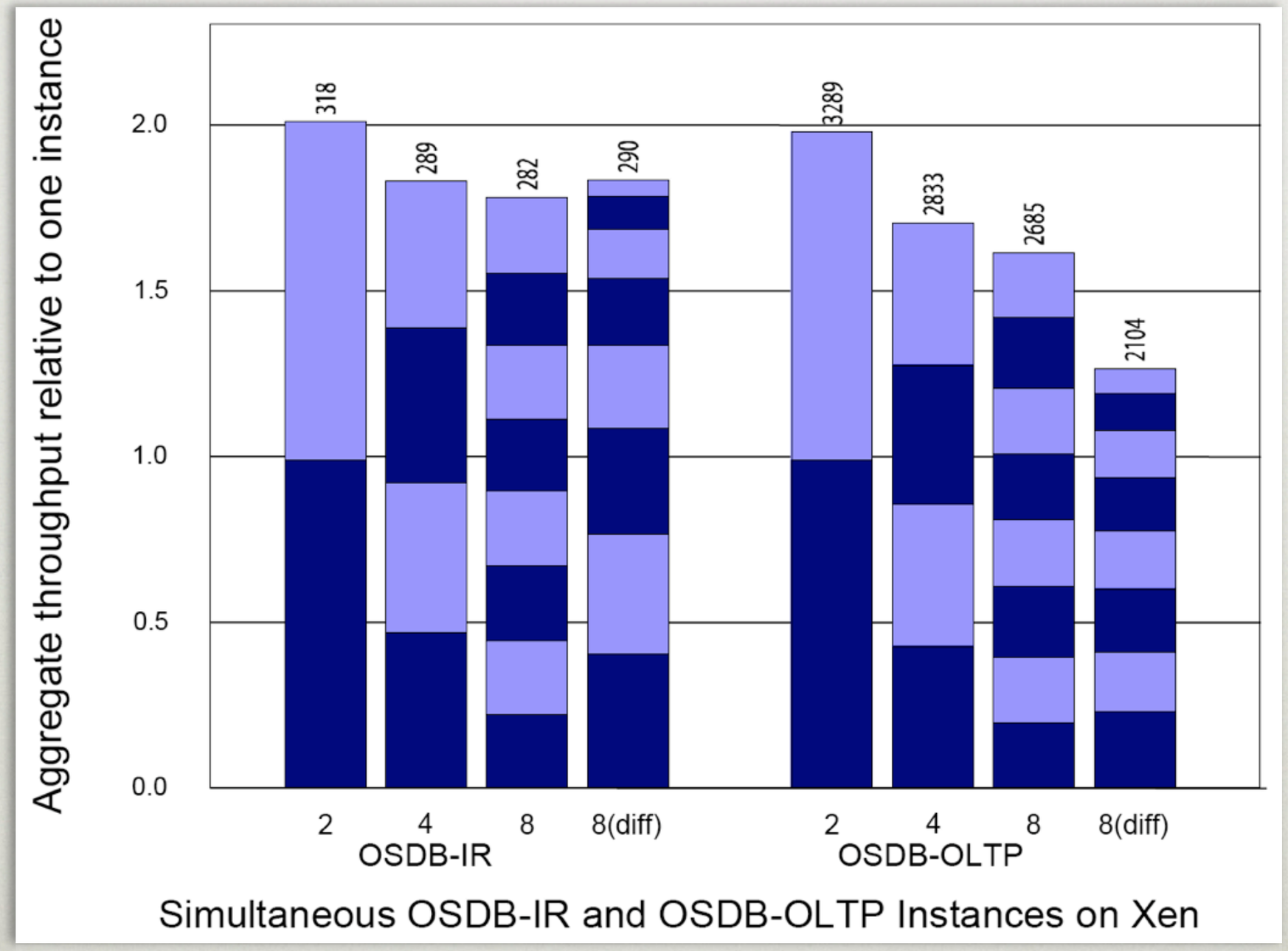
Computationally intensive

SPEC: Standard Performance Evaluation Corporation
OSDB: Open Source Database Benchmark
UML: User-mode Linux
VMware

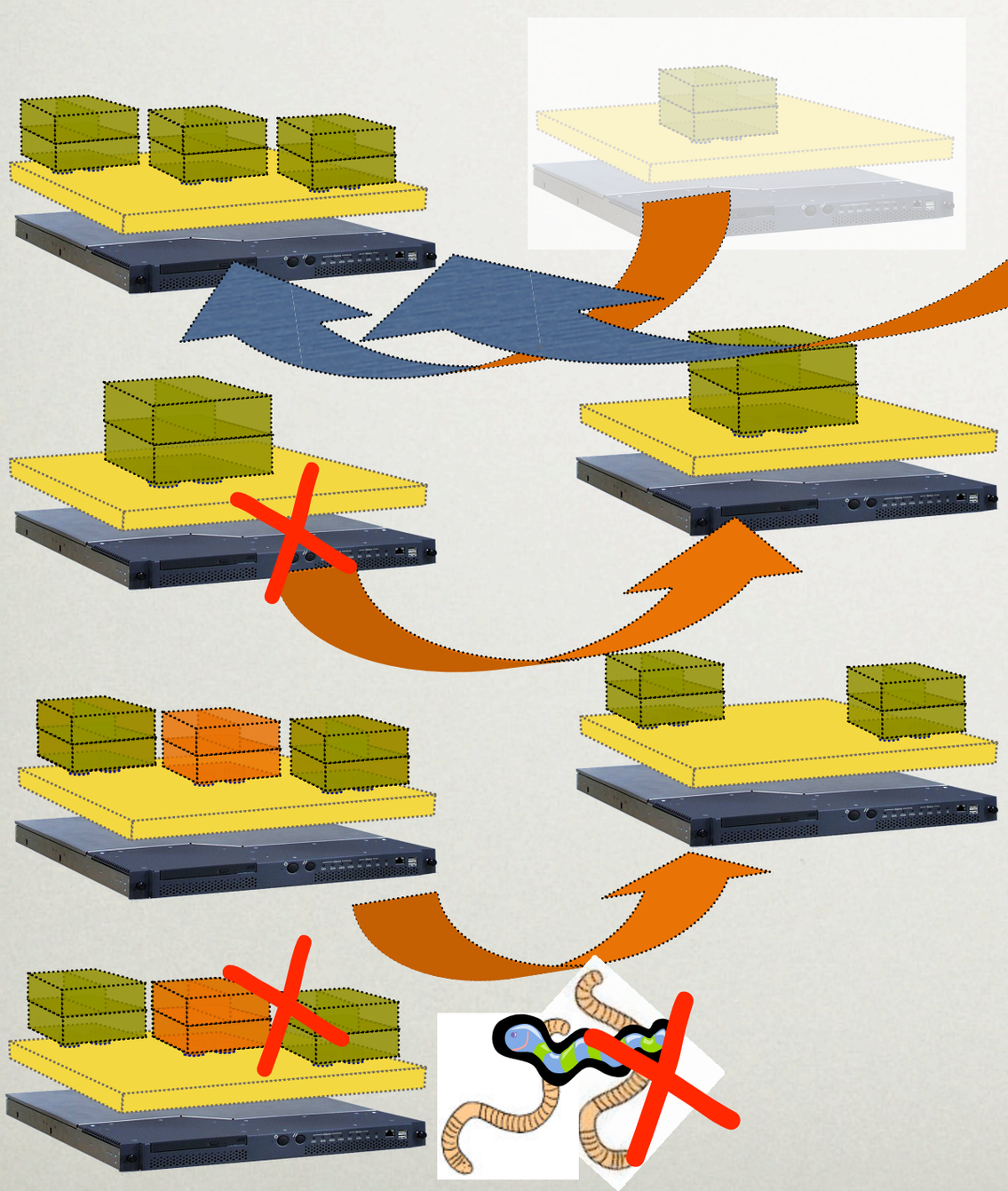
Scalability



Resource Differentiation



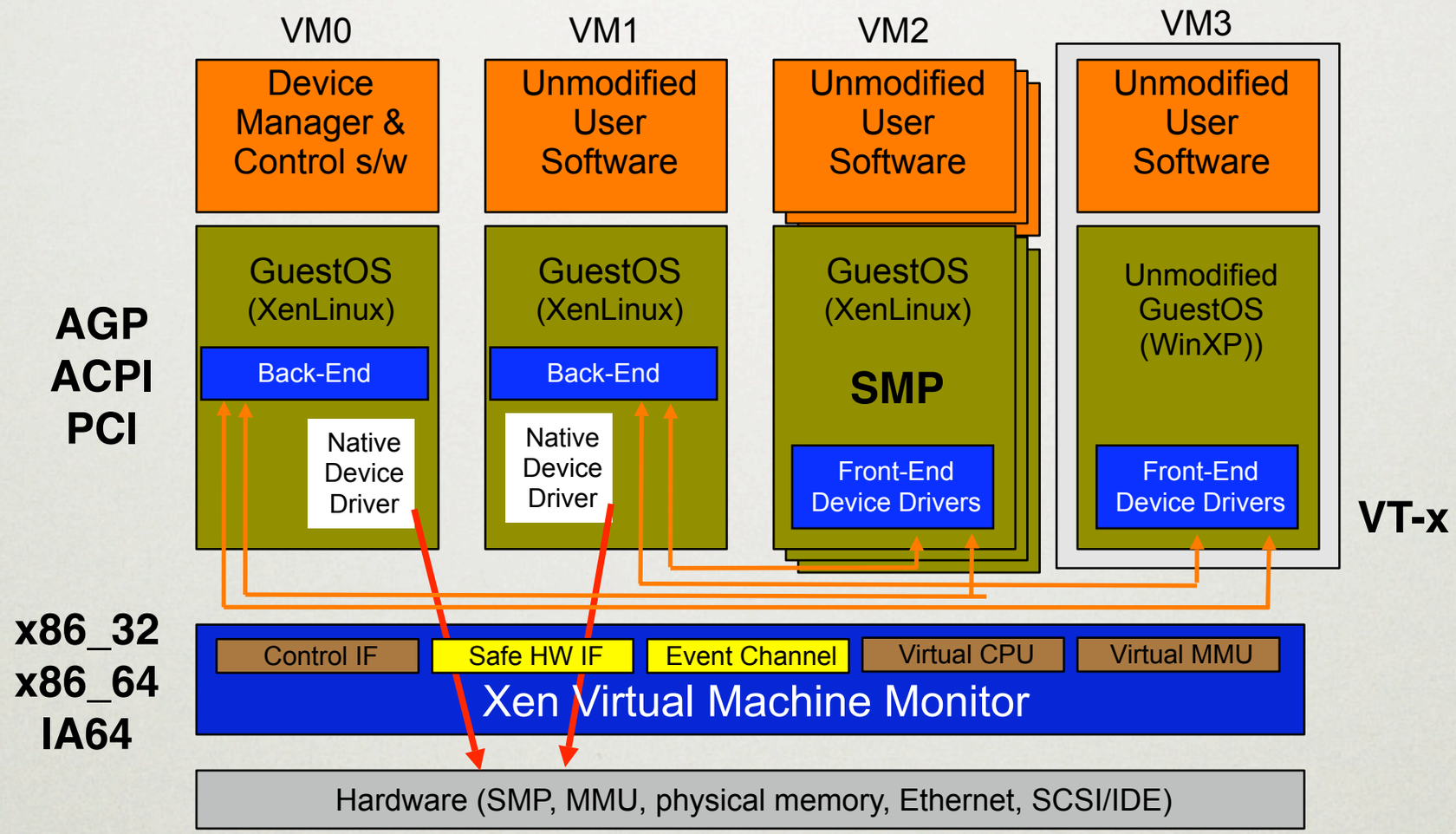
Virtualization in the Enterprise



- Consolidate under-utilized servers to reduce CapEx and OpEx
- Avoid downtime with VM Relocation
- Relocation
- Enforce security policy

Reference: Xen, Ottawa Linux Symposium 2005 presentation

Xen 3.0 Architecture



References

- Ottawa Linux Symposium 2004 Presentation.
- Ottawa Linux Symposium 2005 presentation
- Wikipedia
- Ring Diagrams http://i30www.ira.uka.de/teaching/

Questions

- Xen utilized paravirtualization to improve the performance of VM. Although the performance was improved, OS should be modified. Is there any solution for this inconvenience?
Re-writing some of the OS seems an expensive task.

Questions

- Are these Commodity Guest OS ports realistic? Changing architecture specific code in XP requires XP source correct? Did they have special licensing arrangements with Microsoft to gain access to XP source code? Does this porting effort justify the return on investment?

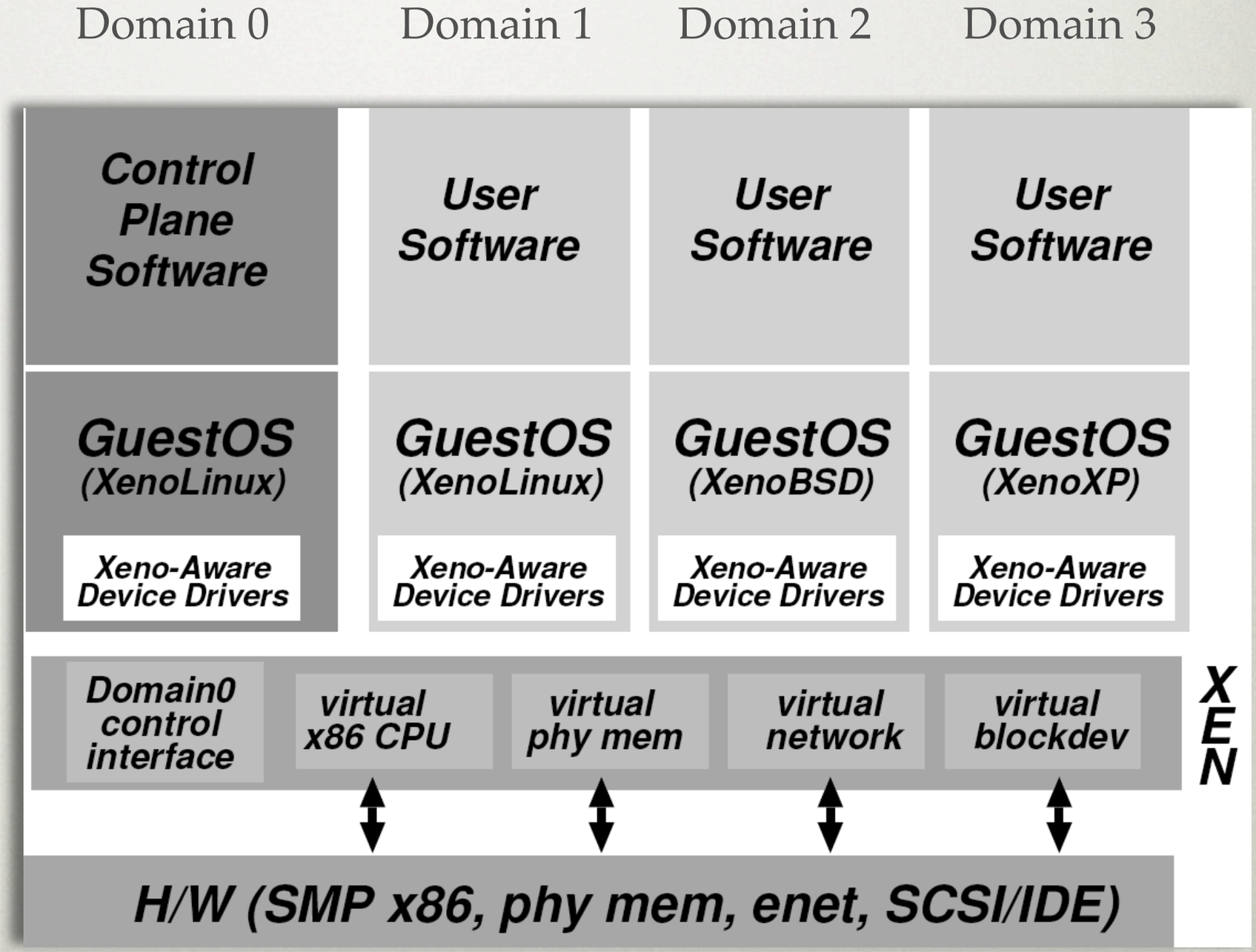
Questions

- Is there any plan to port Xen to other architectures? Is this even possible, or does Xen use some x86 specific features?
- Xen creates a virtual x86 processor. Would it be possible to have Xen create different virtual processors (MIPS, arm, 68k, etc.)?

Questions

- From my understanding, Xen allows a particular guest OS to control resources for all other OS's. Doesn't this introduce more overhead than a lightweight controller built into Xen itself?

Xen 1.2 Architecture



Reference: Barham et al. 2003

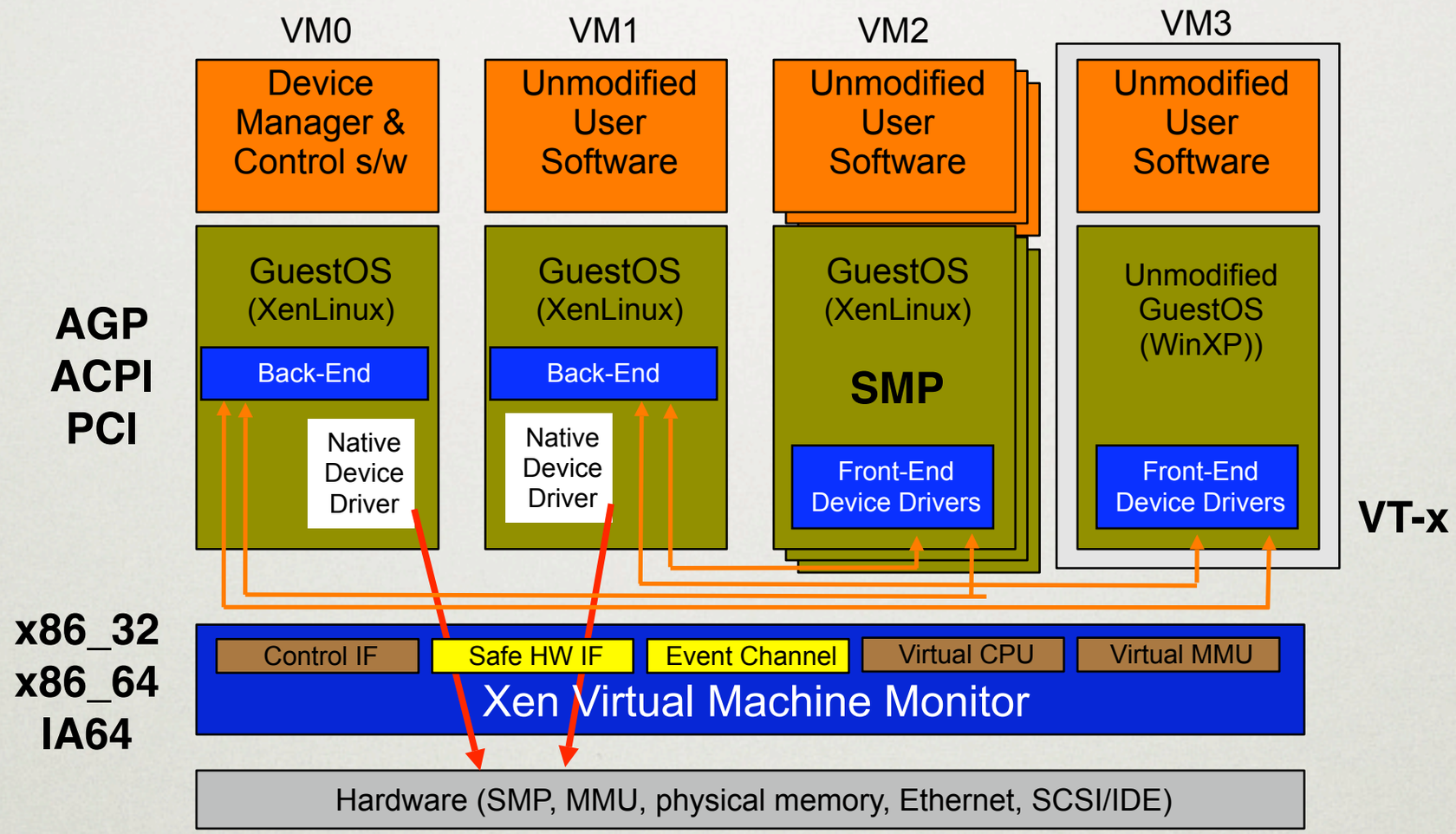
Questions

- How difficult (in your opinion) do you think it is/was to add support for SMP guest OS's?
  - Xen extended to support multiple VCPUs
  - Currently up to 32 VCPUs supported
    - Simple hotplug/unplug of VCPUs

Questions

- Both Intel(IVT) and AMD(AMD-V) added virtualization support to their products. However, the paper doesn't mention how to utilize these virtualization supports at x86 CPU level, why?

Xen 3.0 Architecture



Questions

- In the paper it is mentioned how Xen deals with disk and network I/O. But what about devices that cannot be shared all at once? (e.g. USB Host Controller, Display Adaptor). Do you have any idea if these can be supported in a way or another?

Questions

- I don't feel the author has properly defended that Xen can be extended to run 100 OS's at the same time. Figure 6 shows a slight degredation in throughput, however I don't fully understand it.
- Why does reducing the time-slice degrade performance, and what is this normalized SPEC CINT2000 score?