Xen and the Art of Virtualization

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

University of Cambridge Computer Laboratory

Presented by: Brandon Elting

Outline

- Why Virtualization?
- Overview of Xen
- Benchmark Results
- Xen Today
- Conclusion

Outline

- Why Virtualization?
- Overview of Xen
- Benchmark Results
- Xen Today
- Conclusion

Why Virtualization?

- Problem Domain
- Virtualization
- Paravirtualization

Problem Domain

- Need to execute a diverse range of applications and services
  - Need to support multiple OSes and configurations
  - Solution: Run multiple processes on a single machine
    - Unfortunate configuration interactions
  - Solution: Run separate OSes on different machines
    - Maintenance Issues
    - Keep machines busy
- Support Legacy Software on Modern Hardware

Virtualization

- Present the illusion of many small Virtual Machines to run multiple instances of different Operating Systems concurrently
  - Virtual Machine exactly like physical machine
- Pros
  - Can run unmodified OSes in VM
- Cons
  - Performance
  - Uncooperative hardware
    - Solution: binary rewriting
- Examples
  - VM/370, VMware, Disco

Paravirtualization

- Present an idealized VM abstraction to guest OSes
  - Differs from underlying hardware interface
- Pros
  - Can deal with difficult to virtualize architectures
  - Exposing both a virtual and real interface leads to potential performance enhancements
- Cons
  - Must port existing OSes to run on paravirtualized host
- Example
  - Denali

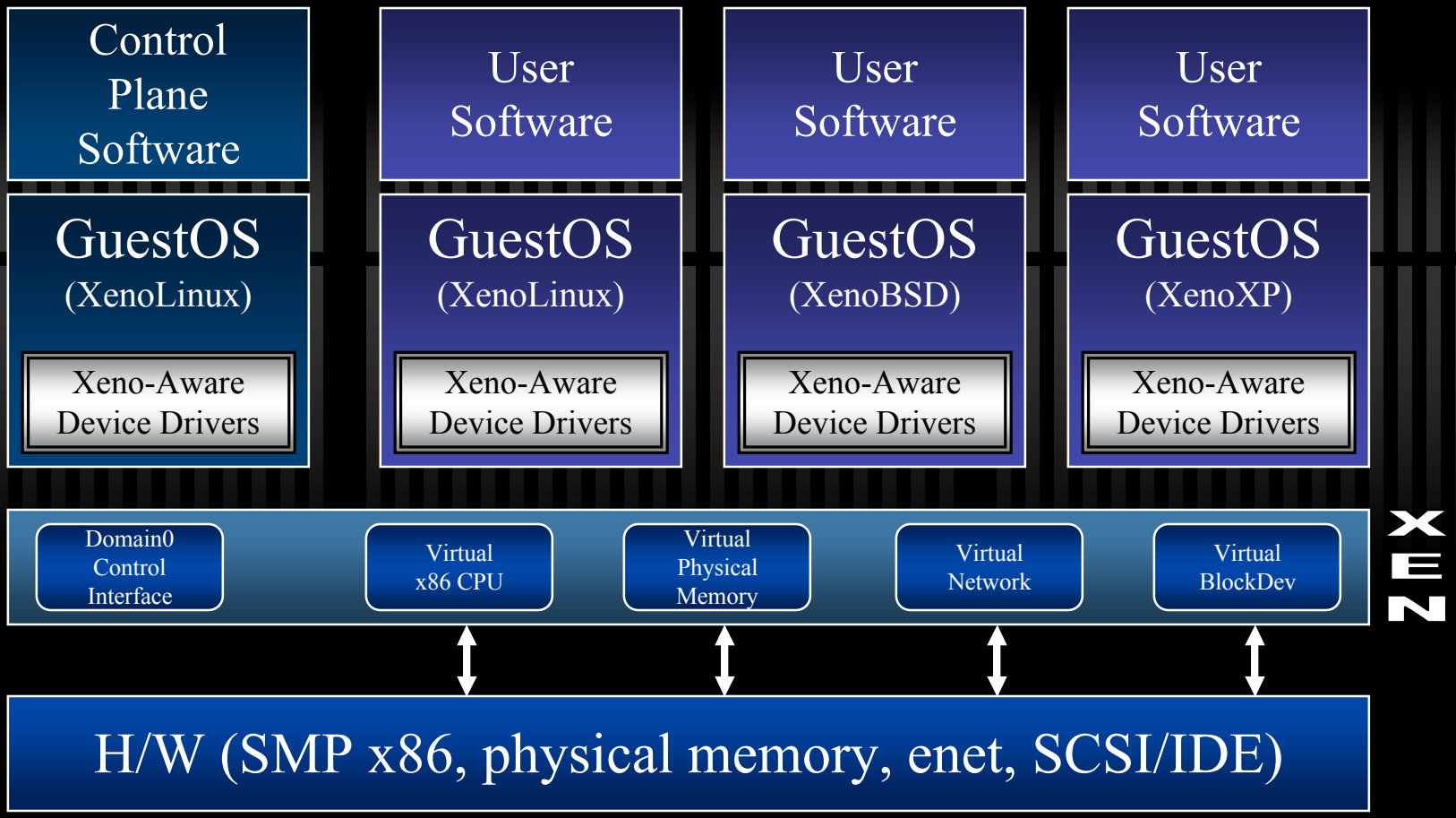
Outline

- ✓ Why Virtualization?
- ⇨ Overview of Xen
- Benchmark Results
- Xen Today
- Conclusion

Overview of Xen

- High-Performance, Paravirtualized Virtual Machine Monitor/Hypervisor
- Runs on 32-bit x86 Processors
- Provides an environment to execute up to 100 VM instances simultaneously
- Focus on Performance Isolation
- Attempts to minimize overhead associated with virtualization
- Supports Ported Guest Operating Systems
  - XenoLinux - Port of Linux 2.4
  - XenoXP - Port of Windows XP (in development)

Machine Running Xen Hypervisor



Virtual Machine Interface

- CPU
- Memory
- Device I/O
- Control Management

Virtualized CPU

- CPU Execution
  - Non-Privileged Instructions can run directly on “bare hardware”
  - Privileged operations must be marshaled by Xen
- Privilege Levels
  - x86 has four privileged levels (rings) 0-3 (0=more … 3=less)
  - Xen runs at ring0, Guest OS at ring1, apps at ring3
- Exception Handling
  - Guest provides virtual IDT
  - Most ISRs same as on real x86 hardware, page-fault is special case
- Interrupts and Events
  - Handled via “event channels,” Xen upcalls into guest
  - Maskable, analogous to disabling interrupts
  - ‘Fast’ exception handler for Guest OS system-calls

Memory Management 1

- Page Tables
  - Guests responsible for managing their own page tables
  - Updates to page tables must go through Xen
  - Guests have direct read access to page tables
  - Updates to page tables can be batched
  - Page frames contain reference count and type
- Translation Lookaside Buffer
  - x86's hardware managed TLB complicates virtualization
  - Xen mapped to the top 64MB of each address space, saves TLB flushes
- Segmentation
  - Handled in a similar manner as page tables

Memory Management 2

- Physical Memory
  - Most Operating Systems expect contiguous memory addresses
  - Illusion of contiguous physical memory provided by physical-hardware map
  - Exposing both hardware and physical memory addresses provides area for optimization
    - Cache locality

Device I/O

- Xen exposes a set of clean and simple device abstractions
- I/O data transferred between guest and Xen via aync I/O Rings
- Network Device
  - VFR - Virtual Firewall-Router
  - VIF - Virtual Network Interface
- Disks
  - VBD - Virtual Block Devices

Control Management

- Domain0
  - Created at Xen boot-time
  - Has access to Xen's control interface
  - Hosts application-level management software
  - Provides separation of policy and mechanism

Control Transfer

- Hypercalls
  - Synchronous communication between Hypervisor and Guest
  - Analogous to system calls
- Events
  - Asynchronous notifications to domains
  - Used to notify domains of device driver interrupts
  - Lightweight notification of important events
    - ie. Domain-termination requests

Data Transfer

- I/O Rings
  - Mechanism to allow efficient moving of data vertically through system
  - Based around two pairs of producer-consumer pointers
  - Unique request ids allow reordering
  - Allows producer to enqueue multiple requests and defer notifying consumer

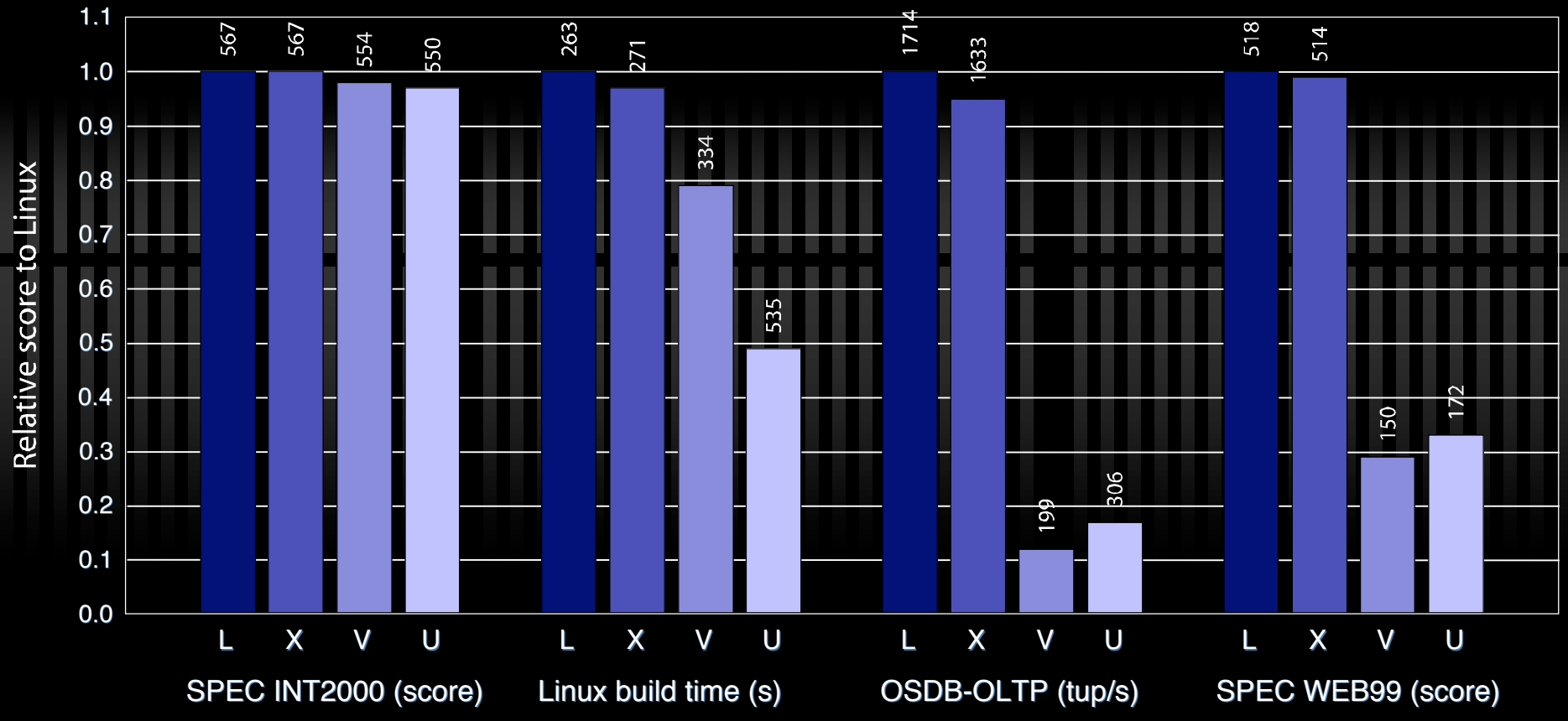
Outline

- ✓ Why Virtualization?
- ✓ Overview of Xen
- ⇨ Benchmark Results
- Xen Today
- Conclusion

Benchmark Results

- Compare Linux to different Virtualization tecnhiques
- XenoLinux compared against
  - Linux, VMWare, User-Mode Linux
- Benchmarks
  - SPEC: cpu intensive
  - Linux build: 7% of time spent in kernel
  - OSDB-IR/OLTP: OS intensive, many domain transitions
  - Dbench: filesystem intensive
  - SPEC WEB99: good overall measure

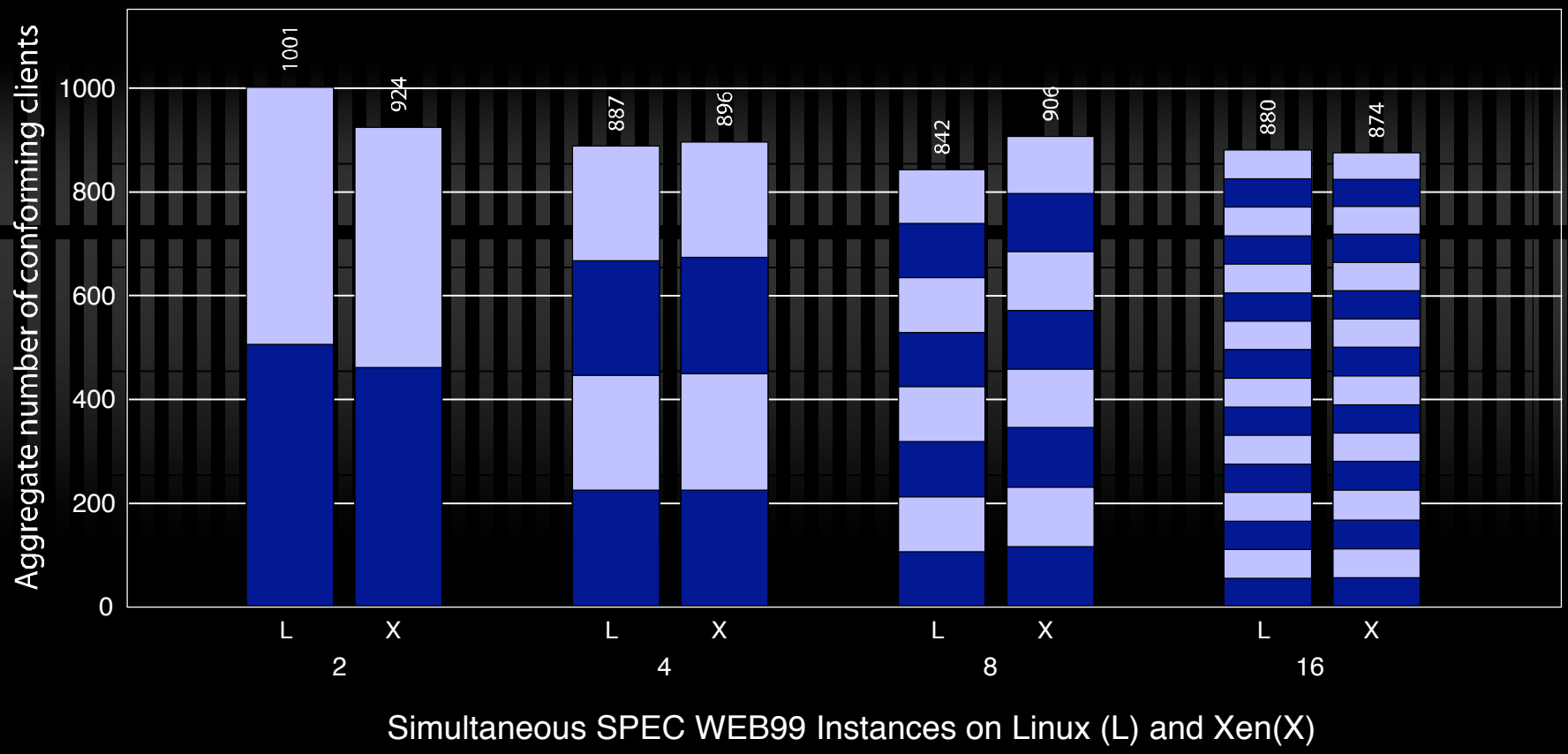
Performance



Benchmark suite running on Linux (L), Xen (X), VMware Workstation (V), and UML (U)

Source: xen-lwe2005-short.ppt

Concurrent VM



Source: xen-lwe2005-short.ppt

Additional Results

- Performance Isolation
  - Execute domains with “anti-social” processes
  - OSDB-IR and SPEC WEB99 only slightly affected
- Scalability
  - Run up to 100 VMs concurrently
  - Only a loss of 7.5% throughput compared to Linux

Outline

- ✓ Why Virtualization?
- ✓ Overview of Xen
- ✓ Architecture
- ✓ Benchmark Results
- ⇨ Xen Today
- • Conclusion

Xen Today

- Current Version: Xen 3.2 (Released Jan ‘08)
- Supports HW Virtualization Extensions (Intel IVT, AMD-V)
  - Runs unmodified OSes
- Supports SMP Virtualized Guest OSes
- Supported OSes: Windows, Linux, Solaris, BSD, …
- Virtualizes Architectures: x86, x86/64, IA64, PowerPC, …
- Live VM Relocation
  - Load balancing across a cluster
- Graphics Virtualization: Direct3D, OpenGL

Outline

- ✓ Why Virtualization?
- ✓ Overview of Xen
- ✓ Benchmark Results
- ✓ Xen Today
- ⇨ Conclusion

Conclusion

- Xen provides excellent platform for deploying a wide variety of differing applications
- Xen provides necessary protection and performance isolation
- Paravirtualization provides near native performance

Outline

- ✓ Why Virtualization?
- ✓ Overview of Xen
- ✓ Benchmark Results
- ✓ Xen Today
- ✓ Conclusion

References

- www.xen.org
- www.cl.cam.ac.uk/research/srg/netos/xen

Questions/Comments?