Xen and the Art of Virtualization

Introduction

- Challenges to build virtual machines
  - Performance isolation
    - Scheduling priority
    - Memory demand
    - Network traffic
    - Disk accesses
  - Support for various OS platforms
  - Small performance overhead

Xen

- Multiplexes resources at the granularity of an entire OS
  - As opposed to process-level multiplexing
  - Price: higher overhead
- Target: 100 virtual OSes per machine

Xen: Approach and Overview

- Conventional approach
  - Full virtualization
    - Cannot access the hardware
    - Problematic for certain privileged instructions (e.g., traps)
    - No real-time guarantees

Xen: Approach and Overview

- Xen: paravirtualization
  - Provides some exposures to the underlying HW
    - Better performance
    - Need modifications to the OS
    - No modifications to applications

Memory Management

- Depending on the hardware supports
  - Software managed TLB
    - Associate address space IDs with TLB tags
    - Allow coexistence of OSes
    - Avoid TLB flushing across OS boundaries

Memory Management

- X86 does not have software managed TLB
  - Xen exists at the top 64MB of every address space
  - Avoid TLB flushing when an guest OS enter/exist Xen
  - Each OS can only map to memory it owns
  - Writes are validated by Xen

CPU

- X86 supports 4 levels of privileges
  - 0 for OS, and 3 for applications
  - Xen downgrades the privilege of OSes
  - System-call and page-fault handlers registered to Xen
  - “fast handlers” for most exceptions, Xen isn’t involved

Device I/O

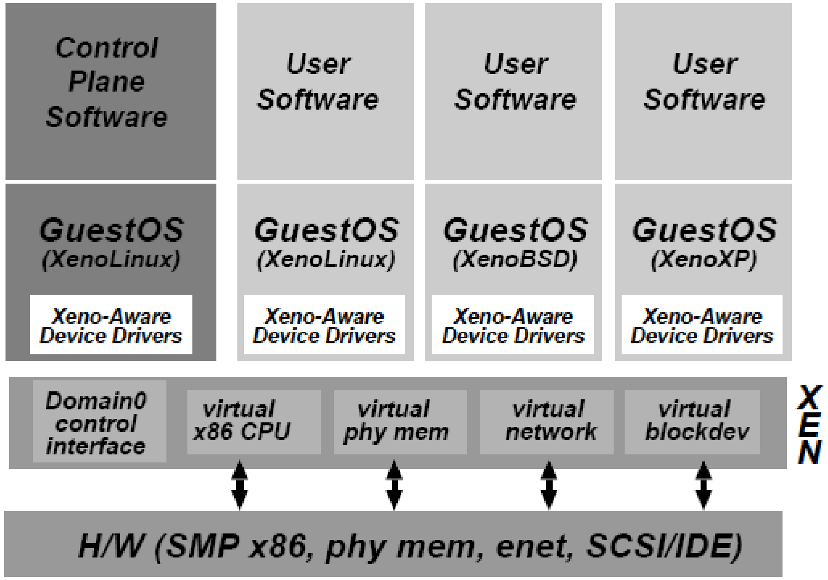
- Xen exposes a set of simple device abstractions

The Cost of Porting an OS to Xen

- Privileged instructions
- Page table access
- Network driver
- Block device driver
- <2% of code-base

Control Management

- Separation of policy and mechanism
- Domain0 hosts the application-level management software
  - Creation and deletion of virtual network interfaces and block devices

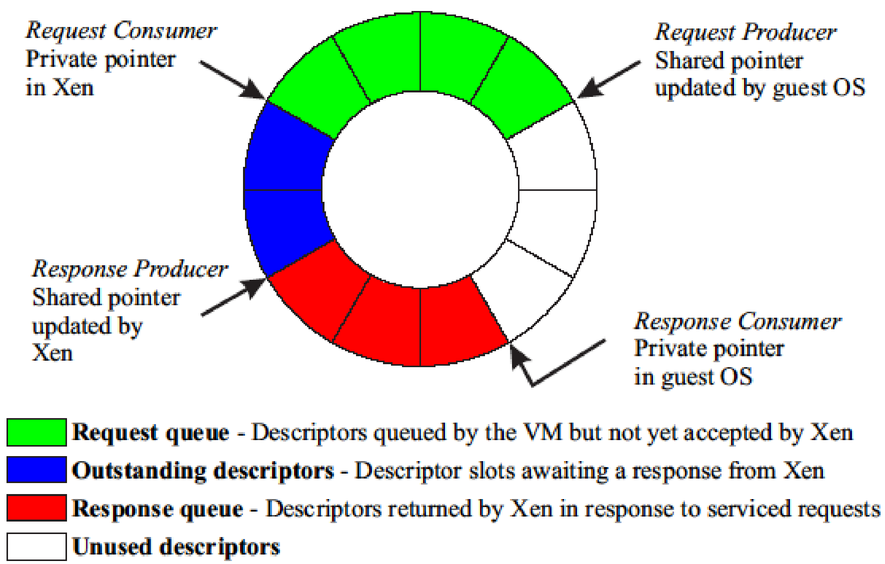


Control Transfer: Hypercalls and Events

- Hypercall: synchronous calls from a domain to Xen
  - Analogous to system calls
- Events: asynchronous notifications from Xen to domains
  - Replace device interrupts

Data Transfer: I/O Rings

- Zero-copy semantics



CPU Scheduling

- Borrowed virtual time scheduling
  - Allows temporary violations of fair sharing to favor recently-woken domains
  - Goal: reduce wake-up latency

Time and Timers

- Xen provides each guest OS with
  - Real time (since machine boot)
  - Virtual time (time spent for execution)
  - Wall-clock time
- Each guest OS can program a pair of alarm timers
  - Real time
  - Virtual time

Virtual Address Translation

- No shadow pages (VMWare)
- Xen provides constrained but direct MMU updates
- All guest OSes have read-only accesses to page tables
- Updates are batched into a single hypercall

Physical Memory

- Reserved at domain creation times
- Memory statically partitioned among domains

Network

- Virtual firewall-router attached to all domains
- Round-robin packet scheduler
- To send a packet, enqueue a buffer descriptor into the transmit rang
- Use scatter-gather DMA (no packet copying)
  - A domain needs to exchange page frame to avoid copying
  - Page-aligned buffering

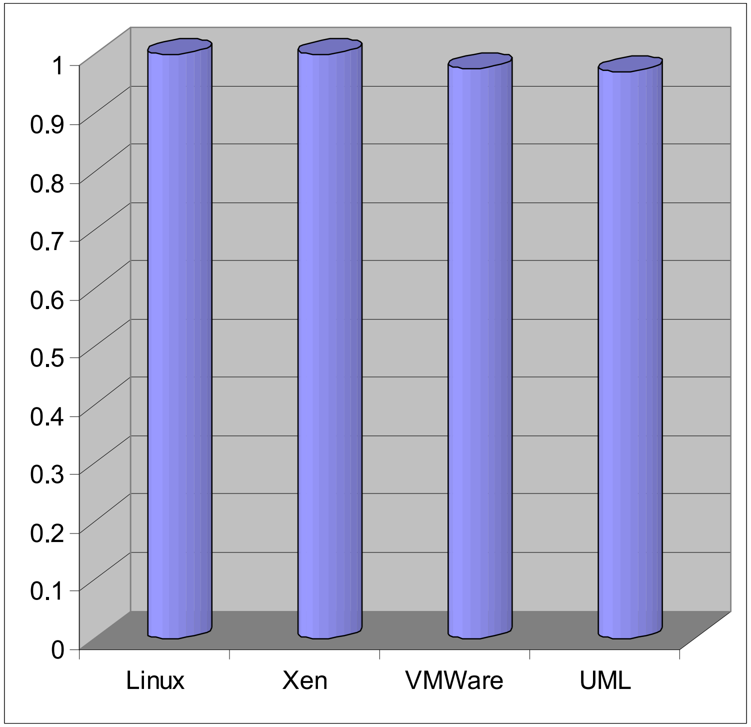
Disk

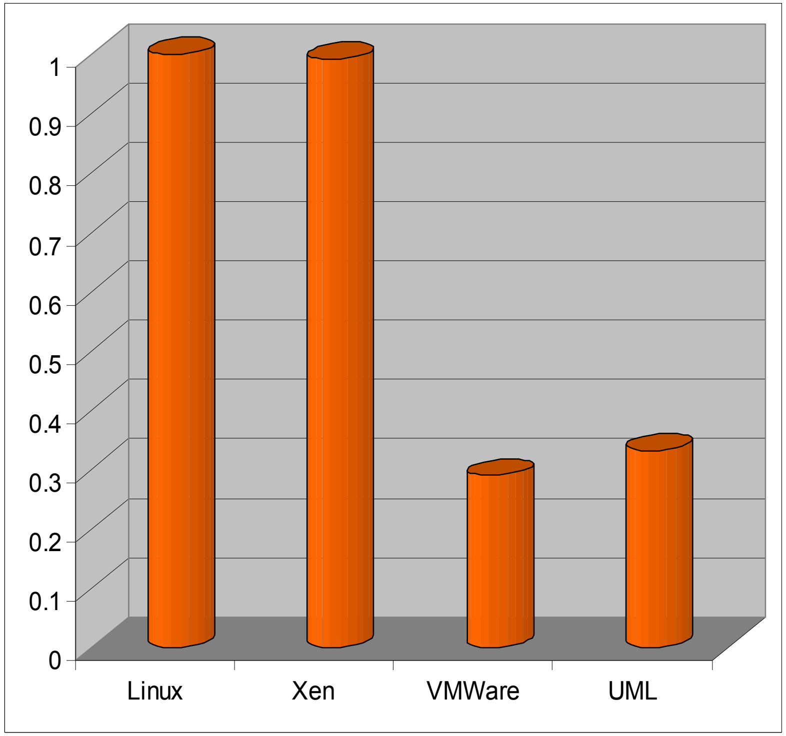
- Only Domain0 has direct access to disks
- Other domains need to use virtual block devices
  - Use the I/O ring
  - Reorder requests prior to enqueuing them on the ring
  - If permitted, Xen will also reorder requests to improve performance
- Use DMA (zero copy)

Evaluation

- Dell 2650 dual processor
- 2.4 GHz Xeon server
- 2GB RAM
- 3 Gb Ethernet NIC
- 1 Hitachi DK32eJ 146 GB 10k RPM SCSI disk
- Linux 2.4.21 (native)

Relative Performance





SPEC INT2000 score

CPU Intensive

Little I/O and OS interaction

SPEC WEB99

180Mb/s TCP traffic

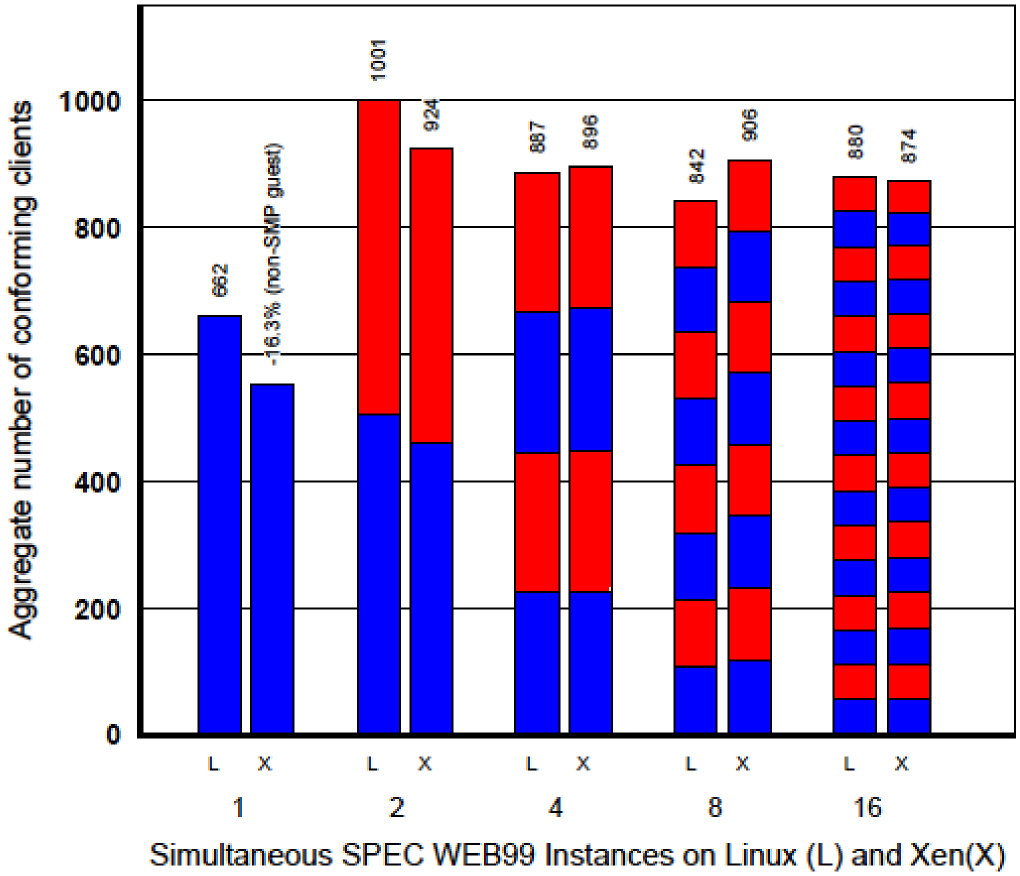
Disk read-write on 2GB dataset

Concurrent Virtual Machines

Multiple Apache processes in Linux

vs.

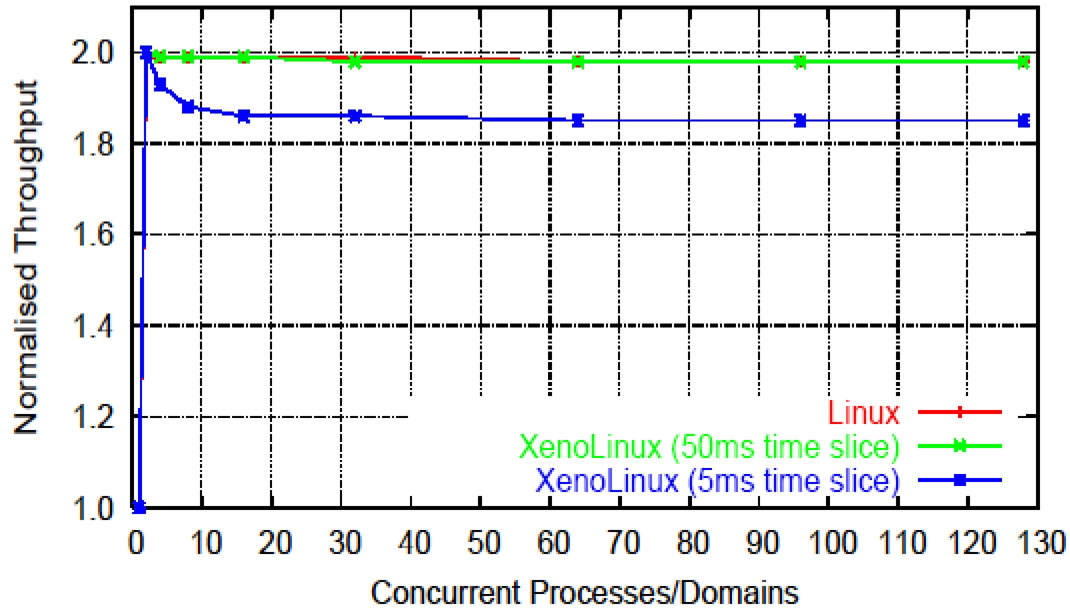
One Apache process in each guest OS



Performance Isolation

- 4 Domains
- 2 running benchmarks
- 1 running dd
- 1 running a fork bomb in the background
- 2 antisocial domains contributed only 4% performance degradation

Scalability



Normalized aggregate performance of a subset of SPEC CINT2000 running concurrently on 1-128 domains