Xen and the Art of Virtualization

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

Outline

- Motivation
- Overview of Xen
- CPU virtualization
- MMU virtualization
- Experimental results
- Recent Developments

Motivation

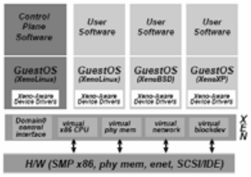
- Stronger isolation between applications
  - Using separate machines is too expensive
  - Separate processes is not sufficient
- Excess computing power
- Different OSs on the same machine

Types of Virtualization

- Hardware-level virtualization
  - Vmware, Xen
- Operating system-level virtualization
  - Jails
- High-level language virtual machines
  - Java VM

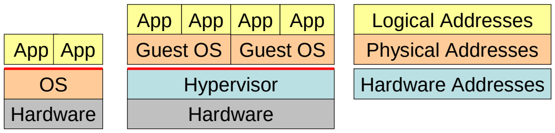
Overview of Xen

- Requires the guest OS to be ported
- Applications run without modifications
- Does not use a host OS



Ideal VM CPU

- Sensitive instructions cause exceptions
  - Instructions that change the machine state
  - Instructions that read or write sensitive registers/memory



App App
OS
Hardware

App App App App
Guest OS Guest OS
Hypervisor
Hardware

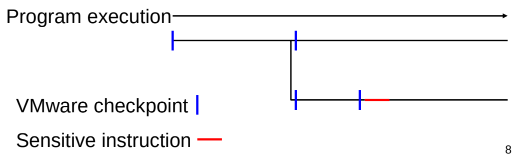
Logical Addresses
Physical Addresses
Hardware Addresses

x86 CPU

- Privileged instructions can only be successfully executed from below the red line
- Some sensitive instructions are not privileged

VMware CPU virtualization

- Checks for sensitive instructions before execution

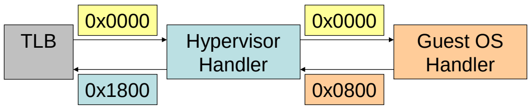


Xen CPU virtualization

- When the guest OS executes privileged instructions, the x86 raises exceptions
- Xen catches these exceptions
- Guest OSs directly call Xen code instead of using sensitive, unprivileged instructions

Ideal VM MMU

- Page translation occurs in software
- OSs provide a TLB miss handler
- Hypervisor executes guest mapping routine



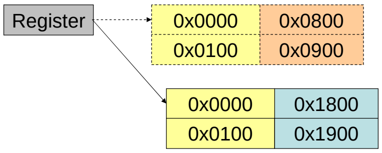
Logical | Physical | Hardware

x86 MMU

- TLB misses are handled directly by the MMU
- OSs must create a page table that maps logical to physical addresses
  - The table must be laid out as specified by the MMU
  - The OS sets a register to point to the table

VMware MMU virtualization

- Maintains shadow page tables



0x0000	0x0800
0x0100	0x0900

0x0000	0x1800
0x0100	0x1900

Logical | Physical | Hardware

Xen MMU virtualization

- Xen exposes the hardware addresses to the guest OS
- The guest OS constructs a page table that maps from logical to hardware addresses
- Updates to the page table must pass through Xen

Experiments

- Minimal performance degradation over plain Linux

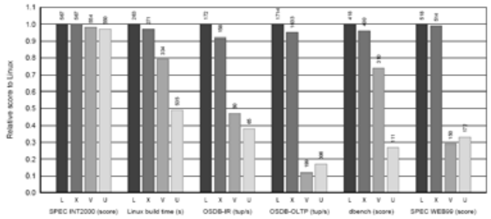
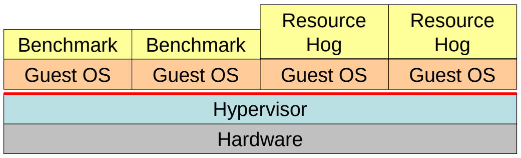


Figure 3: Relative performance of native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

Performance Isolation

- Prevented misbehaving guests from interfering with other guests



Benchmark | Benchmark | Resource Hog | Resource Hog

Guest OS | Guest OS | Guest OS | Guest OS

Hypervisor

Hardware

Recent Developments

- Many Linux distros have Xen support
- Unmodified Windows XP ran on Xen with Intel VT-enabled processors
- Blazingfast provides virtual servers using Xen