Virtualization

Presented by : Walid Budgaga

CS 655 – Advanced Topics in Distributed Systems
Computer Science Department
Colorado State University

1

Outline

- Introduction
- Xen
- Denali
- Comparison
- Conclusion

2

Introduction

3

Virtualization

- Creating a simulated computer environment called virtual machine (VM)
- Enabling running of guest software
  - Users applications
  - Operating systems
- Providing the ability of creating several VMs on one physical machine
  - VMs share physical resources of the host machine

4

Common Terminology

- "Host" is the physical machine that hosts one or more VMs
- "Host OS" is the OS of the host
- "Domain" or "Guest" is a virtual machine
- "Guest OS" is the OS that is running in one domain

5

Why we need virtualization?(1)

- Installing several instances of different OSs on one single machine
  - Developing & running applications for different OSs
- Having the ability to copy VM from one physical machine to another
  - Balance load
  - Improve reliability & availability
  - Recover from failure

6

Why we need virtualization?(2)

- Replacing physical servers by one powerful physical server
  - Increasing the utilization of hardware resources
  - Reducing costs (hardware & energy)
  - Reducing time needed for provisioning new servers
- Using for disaster recovery scenarios
- Faults remain contained within the domain where they occurred

7

Virtualization Techniques

8

Guest Operating System Virtualization

- Host physical machine runs unmodified operating system
- Virtualization application runs on the host OS
  - Starting, stopping and managing each virtual machine
    - Guest OS runs in VM
  - Controlling access to physical hardware resources
- No change required for OSs
- Example:
  - VMware Server & VirtualBox

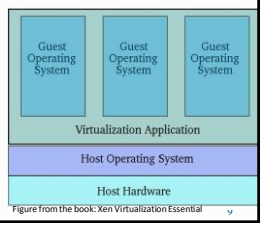


Figure from the book: Xen Virtualization Essential

9

Shared Kernel Virtualization

- Each Guest OS has its own root file system
- All OSs share the same kernel of the host OS
- Example:
  - Linux Vserver
  - FreeVPS

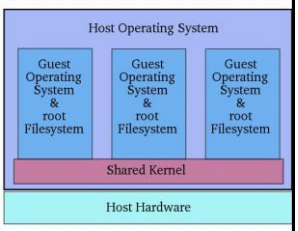


Figure from the book: Xen Virtualization Essential

10

Kernel Level Virtualization

- The host OS has modified kernel that is
  - Designed to manage and control multiple virtual machine
- Each VM owns guest OS
- Each guest OS has its kernel
- Guest OS tightly coupled to the underlying hardware
- Example:
  - User Mode Linux (UML)
  - Kernel-based Virtual Machine (KVM)

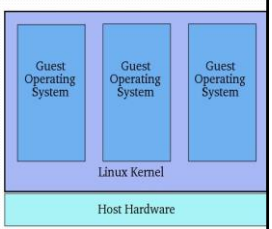
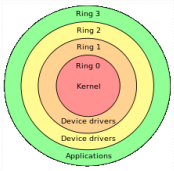


Figure from the book: Xen Virtualization Essential

11

Hypervisor Virtualization (1)

- Privilege levels for X86 CPU are described in rings
  - Ring 0, most privileged level, used by the kernel
  - Ring 1, least privileged level, used by the users applications

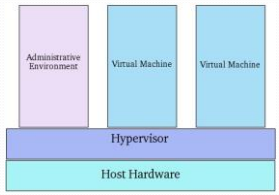


- Hypervisor virtualization
  - Program that is running in ring 0
  - Called type 1 Virtual Machine Monitor (VMM)

12

Hypervisor Virtualization(2)

- Handling resource and memory allocation for VMs
- Providing interfaces for higher level administration and monitoring tools
- Guest OSs must run in less privileged ring



- Problem: OSs designed to run in ring 0

13

Hypervisor: Paravirtualization

- Modifying the kernel of guest OS to run on the hypervisor
- Replacing privileged instructions that run in ring 0 with calls to hypervisor
- Performing sensitive guest OS instructions by hypervisor

14

Hypervisor: Full Virtualization

- Supporting running of unmodified guest OSs on hypervisor
- OSs perform privileged operations as they were running in ring 0
- CPU emulation provided by hypervisor is used
  - To handle and modify privileged operations issued by OS kernels
- Emulation process requires time and resources
  - Increases performance overhead

15

Hypervisor: Hardware Virtualization

- Guest OSs can perform privileged operations in ring 0
- Hypervisor is embedded in the circuits of a hardware component
  - Existing in latest generations of CPUs from both Intel and AMD
- Hypervisor higher privileges levels than OSs

16

17

Overview

- Xen is a paravirtualisation software
- Providing interface to create and manage VMs on single physical machine
- VMs are similar but not identical
- Installing direct on hardware
  - Does not use of host OS
- Targeting to host up to 100 VMs on one physical machine
- Minimum performance overhead

18

Xen VM Interface

19

Memory management

x86 limitations

- x86 does not have a software-managed TLB
  - Missing addresses in TLB requires looking up the page table
- TLB is not tagged
  - Address space switches require complete TLB flush

20

Memory management

Solution

- Guest OSs allocate and manage their own hardware page tables under Xen's control
  - Each OS can only map to its own reserved memory
  - Xen validates write operations
- Xen exists at the top 64MB of every VM's address space
  - Not accessible by domain
  - Avoid to flush TLB when guest OS enters and leaves Xen

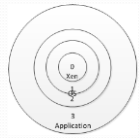
21

CPU

Problem

- X86 provides 4 different privileged levels
  - The highest level is assigned to OS kernel

Solution



- Xen runs at level zero
- OSs can run at level 1 or 2
- Privileged instructions are passed to Xen
- Exceptions are registered with Xen for validation

22

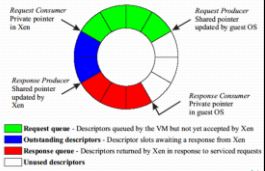
Device I/O

- Set of simple device abstractions are provided
  - To support protection & isolation
- Xen transfers I/O data to and from domain
  - Using shared memory
- Privileged instructions are passed to Xen
- Event mechanism is used to notify domains about hardware interrupts

23

Device I/O Data Transfer(1)

- I/O ring (zero-copy mechanism) is used as shared buffer to exchange data between guest OS & I/O devices through Xen
- The ring is allocated by domain
- The ring buffer does not contain data, but descriptors
- Descriptors reference to data buffer allocated by guest OS



24

Device I/O Data Transfer(2)

- Two pairs of producer & consumer pointers are used to access the ring
- Requests do not need to be processed in order
  - Each request has unique ID
  - Each request reproduced in the associated response
- Each domain can trade-off between latency & throughput
  - By deferring delivery of notifications by specifying minimal number of responses.
  - By invoking a hypercall only after placing number of entries in the ring

25

The Cost of Porting an OS to Xen

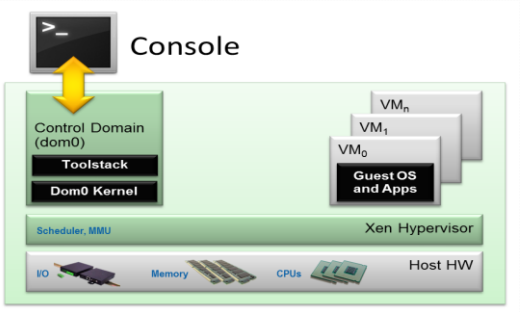
The cost is defined as the number of code lines that need to be added or modified to port OS to Xen

- Linux, XenoLinux, is completely portable
- Window XP, XenoXP, is still in progress
- NetBSD, XenoBSD, is in early stage
-

OS subsection	# lines	
	Linux	XP
Architecture-independent	78	1299
Virtual network driver	484	–
Virtual block-device driver	1070	–
Xen-specific (non-driver)	1363	3321
Total	**2995**	**4620**
(Portion of total x86 code base	1.36%	0.04%)

26

Xen Architecture



27

Control Transfer

Hypercalls

- Synchronous calls from a domain to Xen
- Equivalent to system calls

Events

- Asynchronous event mechanism is used to deliver notifications from Xen to domain
- Virtualizing of device interrupts

28

Scheduling OF VMs

29

Challenges

Virtual machines in a single physical machine

- Host applications with very different timing requirements
  - Sometimes conflicting
- Must safely coexist
- Must work independently and cooperatively

30

VM Scheduler

- Virtual machine monitor that manages VMs switches between their applications
  - Switching between VMs is switching between process
- VM can schedule its own processes
  - But only relative to its own assigned time

31

VM Scheduler Requirements

- Deterministic mapping between real time and virtual time.
  - For time driven processes
  - Time should be known in advance
- Dynamic reallocation of spare computing time
  - To improve resource utilization
- Since each VM could have real & none-real time processes
  - VM is a good place to use dynamic reallocation strategy
- Distinguish between virtual and real time
  - For time sensitive tasks

32

VM Scheduling in Xen (1)

Borrowed virtual time scheduling

- Proportional share scheduler
- Allowing of borrowing processor allocation for future use
- Allowing the assignment of weights to each individual VM
- According to weights processing capacity is relatively partitioned to VMs

33

VM Scheduling in Xen (2)

Borrowed virtual time scheduling

- Used to balance load
- Reduces scheduling latencies
- Temporarily violates fair sharing to favor recently-woken domains.
  - They can borrow time that they have to pay later

34

Time & Timer

- 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

35

Evaluation

36

Experiments Setup

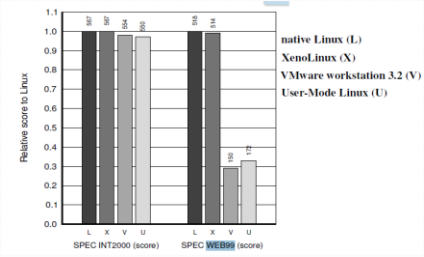
All experiments are performed using

- Dell 2650 dual processor 2.4GHz Xen server
- 2GB RAM
- Broadcom Tigon 3 Gigabit Ethernet NIC
- Single Hitachi DK32EJ 146GB 10k RPM SCSI disk
- Linux 2.4.21 (native)

37

Relative Performance

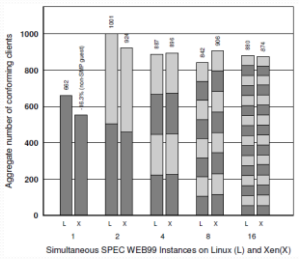
- Evaluation of overhead of the various virtualization techniques relative to running on the `bare metal'.



38

Concurrent Virtual Machines

Results of running 1, 2, 4, 8 and 16 copies of SPEC WEB99 benchmark in parallel on native linux and Xen



39

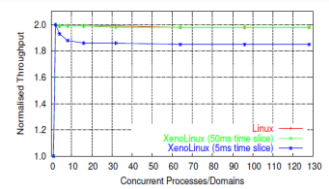
Performance Isolation

- 4 domains configured with equal resource allocations
- 2 running benchmarks(PostgreSQL/OSDB, SEP WEB99)
- 1 running disk bandwidth
- 1 running “fork bomb”

- **Result**
- 2 benchmark domains showed only 4% & 2% performance degradation

40

Scalability



41