DREAM:Lab

SE252:Lecture 4, Jan 20

ILO2:Cloud Virtualization, *Abstractions and Enabling Technologies*

Yogesh Simmhan

Ongoing Assignments

Project Preliminaries

- Project 0, due on Thu Jan 22, midnight

Reading Assignment

- Textbook
  - Chapters 1.3 – 1.4
  - Chapters 4.1 – 4.4

Misc Tasks

- Sign up on mailing list http://mailman.serc.iisc.in/mailman/listinfo/se252.jan15

DREAM:Lab

Lecture 4

Virtualization

ILO 2: Cloud Virtualization, Abstractions & Enabling Technologies

- *You should be able to*
  - *Explain* virtualization and their role in elastic computing.
  - *Describe* service oriented architectures that are foundational to the WWW. ✓
  - *Characterize* the distinctions between Infrastructure, Platform and Software as a Service (IaaS, PaaS, SaaS) abstractions, and Public and Private Clouds, and
  - *analyse* their advantages and disadvantages.

Why Virtualize?

- Share same hardware among independent users
  - Degrees of HW parallelism increasing
- Reduce HW foot print thru' consolidation
  - Eases management, energy usage
- Sandbox/migrate applications
  - Flexible allocation, utilization
- Decouple applications from underlying HW
  - Allows HW upgrades without impact on OS image

Virtualization raises the Abstraction

?

- Similar to *Virtual Memory* to access larger address space
  - *Physical memory* mapping is hidden by OS using *paging*
- Similar to hardware emulators
  - Allow code on one arch to run on a different
- Physical devices -> Virtual Devices
  - CPU, Memory, VHD, NIC
- Now worry/not be aware of physical hardware details

Virtualization Requirements*

- Efficiency Property
  - *All innocuous instructions are executed by hardware*
- Resource Control Property
  - *It must be impossible for programs to directly affect system resources*
- Equivalence Property
  - *A program with a VMM performs in a manner indistinguishable from another without*
  - *Except: (1) Timing, (2) Resource Availability*

* Formal Requirements for Virtualizable 3rd Generation Architectures, Popek & Goldberg, *CACM*, 1974

Types of Virtualization

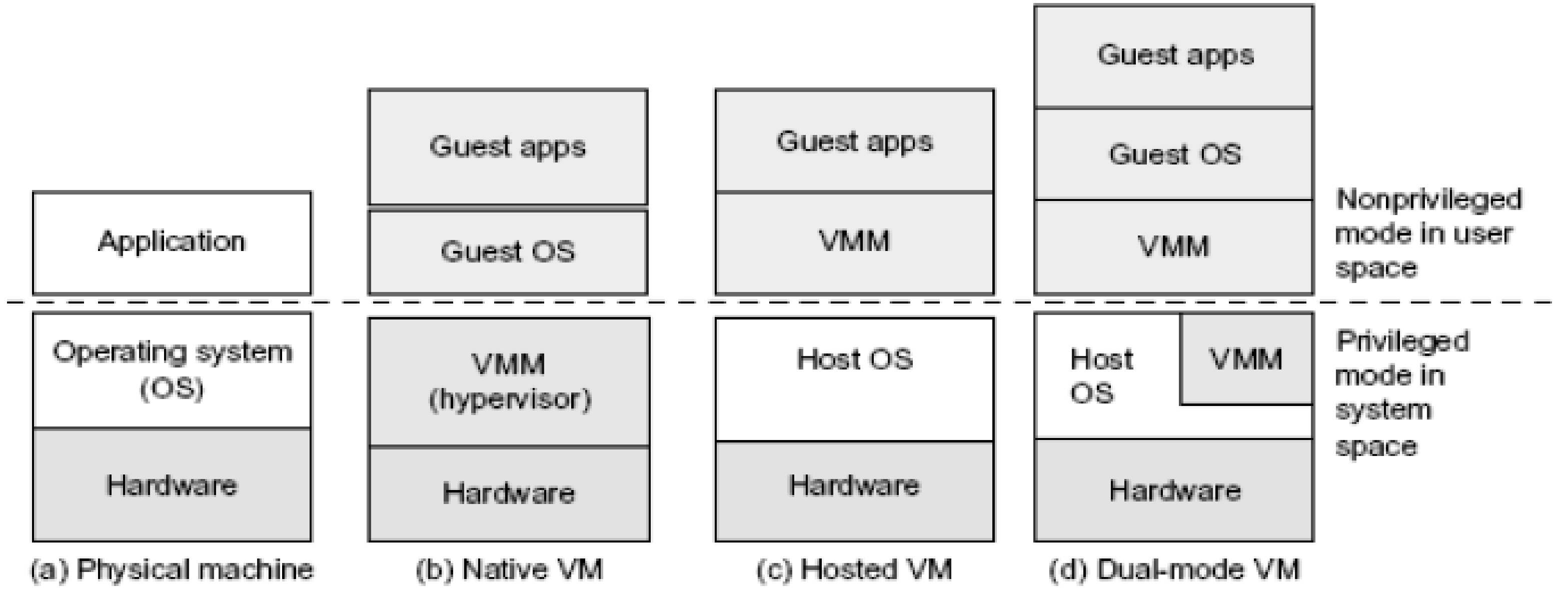


FIGURE 1.12

Three VM architectures in (b), (c), and (d), compared with the traditional physical machine shown in (a).

- Virtual Machine Manager/Virtual Machine Monitor/Hypervisor … *a Caretaker*
- Native (Hyper-V, ~KVM), Hosted (Xen)

Fig. from "Distributed & Cloud Computing, Hwang, et al, MKP; *Courtesy of Kai Hwang, USC*

Types of Virtualization*

- Full Virtualization
  - *Unmodified* Guest OS
  - VMM *binary translates kernel* to *trap* privileged calls
  - Software emulation
  - VMWare Server, Apple Parallels
- Pros
  - Guest OS not modified
  - No HW support required
- Cons
  - Binary translation costly, difficult

?

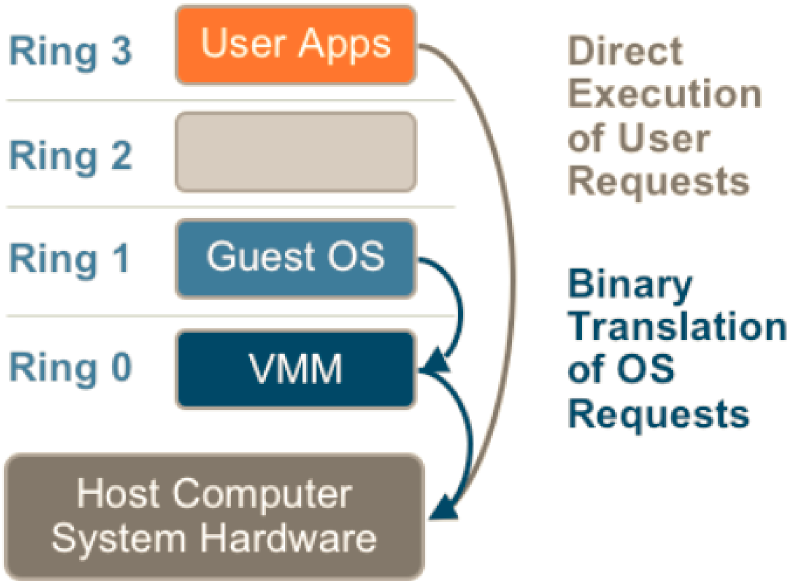


Figure 5 – The binary translation approach to x86 virtualization

*Understanding Full Virtualization, Paravirtualization and Hardware Assist, VMWare, Tech Report WP-028-PRD-01-01, 2007

Types of Virtualization

- Para-virtualization
  - Guest OS *modified* to make "hyper-calls" for privileged instructions
  - Xen in para mode
- Pro
  - (Mostly) faster & easier than bin. translation
- Con
  - Guest OS modified… Legacy, maintenance

?

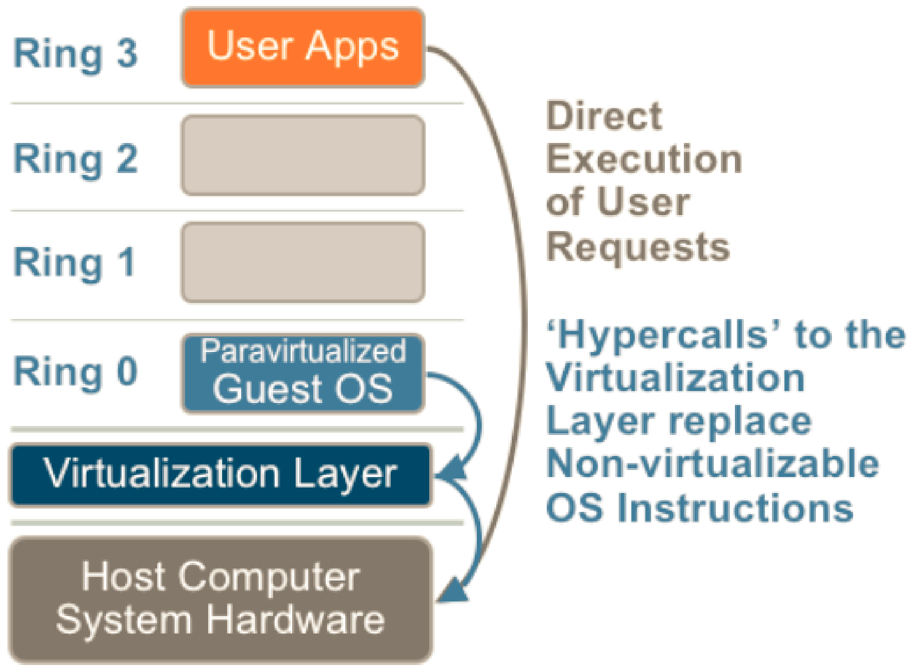


Figure 6 – The Paravirtualization approach to x86 Virtualization

Types of Virtualization

- H/W Assisted Virtualization
  - *Unmodified* Guest OS
  - CPU traps & calls VMM for privileged calls
  - CPU support in Intel VTx, AMD-V
  - Xen HVM, Hyper-V, KVM
- Pros
  - Faster to execute
  - Easier management
- Cons
  - Requires CPU support

?

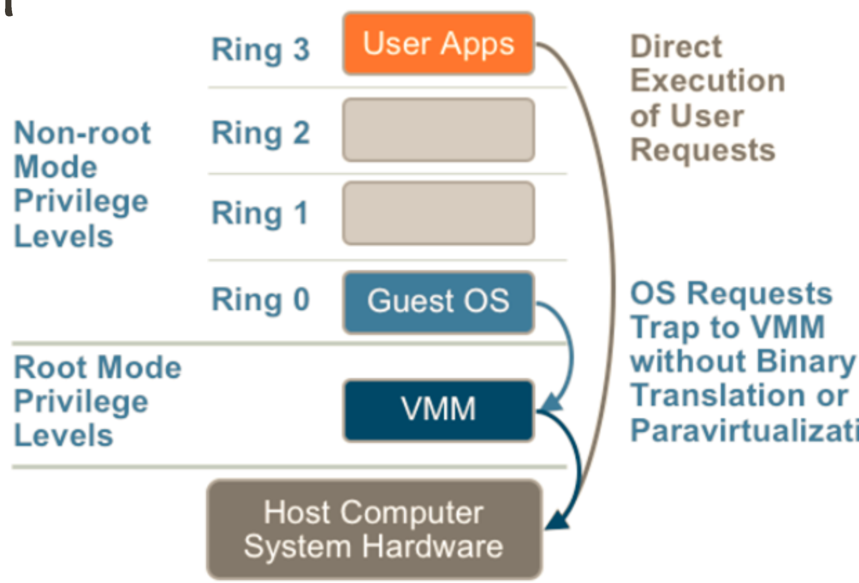


Figure 7 – The hardware assist approach to x86 virtualization

Types of Virtualization

- OS Level Virtualization
  - OS provides containers for isolation
  - Retains host OS image
  - chroot, Docker,LXC (LinuX Containers)
- Pros
  - Faster to boot
  - Fewer images to maintain
  - No CPU support required
- Cons
  - Virtual "Environment"
  - No guest OS, limited distros

?

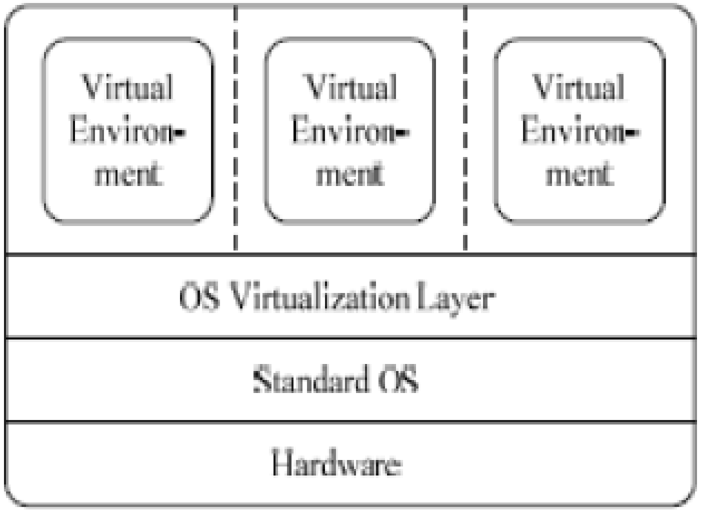
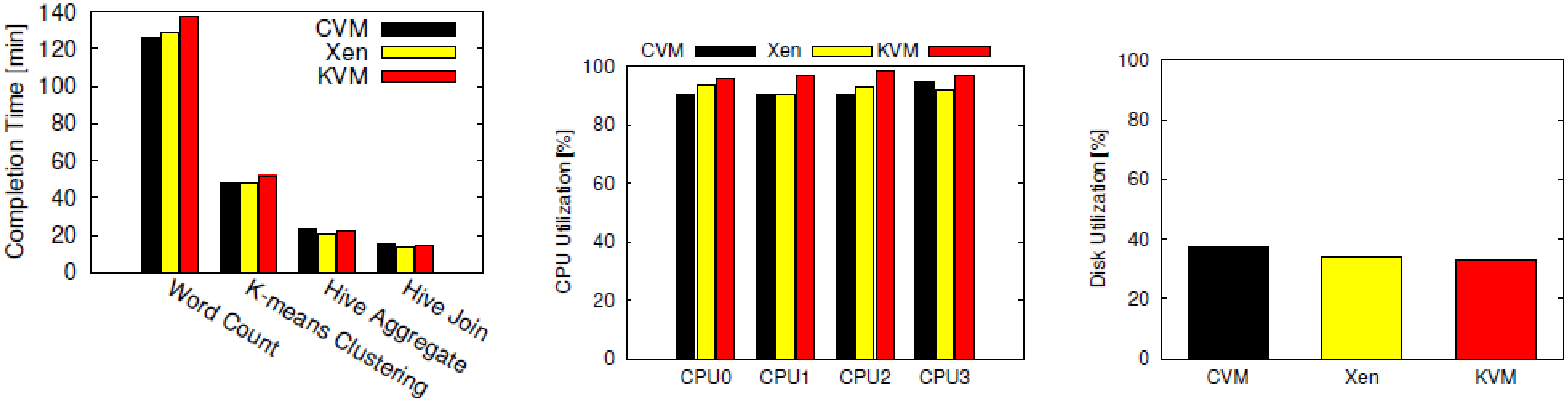


Fig. from "Distributed & Cloud Computing, Hwang, et al, MKP; *Courtesy of Kai Hwang, USC*
See http://www.scriptrock.com/articles/docker-vs-lxc

Performance Effect of HyperVisors



(a) CPU-Bound Benchmark Completion Times. The performance difference is small.

(b) Word Count Average CPU at VM Level. The CPU for each VM is heavily saturated.

(c) Word Count Average Disk Utilization at VM Level. The disk utilization for the VM is low.

Fig. 1: CPU-Bound Benchmark Results and Word Count Statistics. The performance difference for these benchmarks as seen in Figure 1(a) is negligible between the different hypervisors. A representative benchmark, Wordcount, shows high CPU utilization and low disk utilization during the job as seen in Figure 1(b) and 1(c).

* Performance Overhead Among Three Hypervisors: An Experimental Study using Hadoop Benchmarks, Jack Li, Qingyang Wang, Deepal Jayasinghe, Junhee Park, Tao Zhu, Calton Pu, *IEEE Big Data Congress*, 2013

Performance Effect of HyperVisors

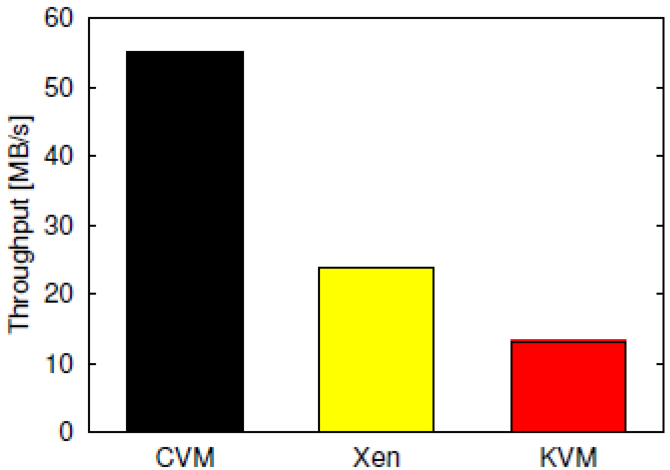
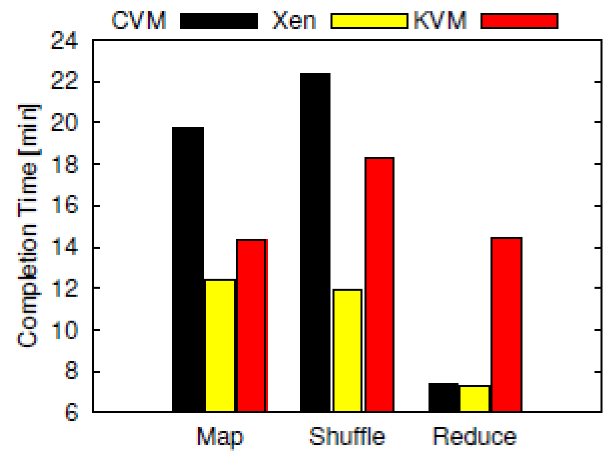


Fig. 3: Filebench Write Microbenchmark. The write microbenchmark results support the disk write throughput trend seen in TestDFSIO Write.



(c) 10GB TeraSort Duration of Each Phase. CVM spends most of the time in the map (heavy in disk reads) and shuffle (heavy in both disk reads and writes). KVM spends a lot of the time during shuffle and reduce (heavy in disk writes).

* Performance Overhead Among Three Hypervisors: An Experimental Study using Hadoop Benchmarks, Jack Li, Qingyang Wang, Deepal Jayasinghe, Junhee Park, Tao Zhu, Calton Pu, *IEEE Big Data Congress*, 2013

Reading from Today's Lecture

- Textbook, Sec 3.0 – 3.3
- Understanding Full Virtualization, Paravirtualization and Hardware Assist, *VMWare, Tech Report WP-028-PRD-01-01*, 2007
- Formal Requirements for Virtualizable 3rd Generation Architectures, Popek & Goldberg, CACM, 1974
- Performance Overhead Among Three Hypervisors: An Experimental Study using Hadoop Benchmarks, Jack Li, Qingyang Wang, Deepal Jayasinghe, Junhee Park, Tao Zhu, Calton Pu, IEEE Big Data Congress, 2013