④ Production:

- app^{n} is accessible to end users under agreed SLA.
- On customer demand; ~~we can~~ provider can include new terms & conditions

⑤ Termination:

- When the customer decides to withdraw hosted application; all related data is transferred to customer, except for legal data.

UNIT-II

VIRTUAL MACHINES & VIRTUALIZATION OF CLUSTERS AND DATA CENTERS:

→ Implementation Levels of Virtualization:

* Levels of Virtualization:

→ Virtualization is a computer technology by which multiple virtual machines (VMs) are multiplexed in the same hardware.

* Purpose of VM:

- enhance resource sharing
- improve computer performance.

** Idea of virtualization is separate hardware and software to yield better system efficiency.

** A traditional computer runs with a host OS specially tailored for its architecture

After virtualization, different user applications managed by their own OS/ guest OS can run on same hardware independent of host OS. This is done by adding additional software, called a virtualization ~~order~~ layer, known as hypervisor or VMM (Virtual Machine Monitor)

** Virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system.

* Common virtualization layers include:

** Instruction Set:

Instruction set emulation leads to

virtual ISAs created on any hardware machine.

→ Major emulation depends on code interpretation

→ For better interpretation process; we use dynamic binary translation.

↳ translates basic blocks of dynamic source instructions to target instructions.

*→ Hardware Abstraction Level:

→ Generates a virtual hardware environment for a VM & process manages underlying hardware through virtualization.

→ Idea is to virtualize computer resources (processors, memory & I/O devices) and upgrade utilization rate by multiple users concurrently.

*→ Operating System Level:

→ abstraction layer b/w traditional OS and user applications.

→ creates isolated containers on a single physical ~~layer~~ server & OS instances to utilize the hardware & software in data centers.

→ This virtualization is used:

- to create virtual hosting environments
- to allocate hardware resources among a large no. of mutually distrusting users.

*→ Library Support Level:

→ Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS.

→ possible by controlling the communication link b/w applications and the rest of system.

*→ User-Application Level:

→ virtualizes an application as a VM.

→ application level virtualization is also known as process-level virtualization ∵ application runs a process.

→ Virtualization layer sits as an application program on top of the OS and exports an abstraction of a VM that can run

programs written and compiled to a particular abstract machine.

→ Application isolation is wrapping the application in a layer that is isolated from the host OS and other applications.

* VMM Design Requirements and Providers:

→ The layer between real hardware & traditional OS is called Virtual Machine Monitor (VMM) and it manages the hardware resources of a computing system.

→ Several traditional OS can sit on the same set of hardware simultaneously.

*→ Requirements of a VMM:

→ Provide an environment for programs - identical to the original machine.

→ Programs run in this environment shd show only minor decreases in speed.

→ complete control of the system resources.

* Exceptions:

→ differences caused by availability of system resources

→ differences caused by timing dependencies.

* Control of resources by a VMM:

→ VMM is responsible for allocating hardware resources for programs.

→ It is not possible for a program to access any resource not explicitly allocated to it.

→ It is possible under certain circumstances for a VMM to regain control of resources already allocated.

* Virtualization Support at the OS level:

→ Cloud computing has 2 challenges:

*→ Ability to use a variable number of physical machines and VM instances depending on the needs of a problem.

*→ Concerns about the slow operation of instantiating new VMs.

*→ OS level virtualization provides a feasible sol^n for these hardware-level virtualization

issues.

→ OS virtualization (also called single-OS image virtualization) inserts a virtualization layer inside an OS to partition a machine's physical resources.

→ This enables multiple isolated VMs within a single OS kernel. This kind of VM is often called virtual execution environment (VE), Virtual Private System (VPS) (or) simply container.

* Advantages of OS extensions:

① VMs at OS level have minimal startups, low resource requirement & high scalability

② It is possible for a VM & its host environment to synchronize state changes when necessary.

→ These are achieved by 2 mechanisms:

① All OS level VMs on the same physical machine share a single OS kernel.

② The virtualization layer can be designed in a way that allows processes in VMs to access as many resources of host machine as possible.

* Disadvantages of OS extensions

→ all VMs at OS level on a single container must have same kind of guest OS.

→ the access requests from a VM need to be redirected to the VM's local resource partition on the physical machine.

* Virtualization on Linux (or) Windows Platforms:

→ Linux kernel offers an abstraction layer - to allow software processes to work with & operate on resources without knowing the hardware details. ⇒ Linux vServer & OpenVz.

→ FVM is an attempt specifically developed for virtualization on the Windows NT.

* Middleware Support for Virtualization:

→ Library level virtualization is also known as user-level Application Binary Interface (ABI) (or) API emulation.

→ This type of virtualization can create execution environments for running alien programs on a platform.

* Virtualization for data center automation:

→ Data center automation means that huge volumes of hardware, software and database resources in these data centers can be allocated dynamically to millions of Internet users simultaneously, with guaranteed QoS and cost-effectiveness.

→ Server Consdidation in data centers:

* In data centers, a large no. of heterogeneous workloads can run on servers at various times.

* Heterogeneous Work loads:

→ Chatty workloads: may burst at some point & return to a silent state at some other point.
Ex: Web Video Service.

→ Non-interactive workloads: do not require people's effort to make progress after they are submitted.
Ex: High-performance computing.

→ In a data centre, most servers are under utilized. A large amount of hardware, space, power and management of these servers is wasted.

→ Server consolidation is an approach to improve the low utility ratio of hardware resources by reducing no. of physical servers.

• Virtualization based server consolidation is the powerful consdidation technique.

→ Server virtualization drawbacks/side effects:

1. Consolidation enhances hardware utilization
2. Enables more agile provisioning and deployment of resources.
3. Total cost of ownership is reduced.
4. Improves availability and business continuity

* Virtual Storage Management:

→ Virtual storage includes the storage managed by VMMs and guest OSes.

* Data stored is classified into 2 types:
  (1) VM images: special to the virtual environment
  (2) Application data: all other data which is same as the data in traditional OS environments.

* Important aspects of System virtualization are encapsulation and isolation.

→ To achieve encapsulation and isolation, both system software and the hardware platform, such as CPUs and chipsets are rapidly updated.

→ Storage Management of underlying VMM is much more complex than that of guest OSes.
  • Operations such as remapping volumes across hosts and check pointing disks are frequently clumsy and sometimes simply unavailable.

→ Content Addressable Storage (CAS) is a solution to reduce the total size of VM images and therefore supports a large set of VM based systems in data centers.

* Cloud OS for virtualized data centers

→ Data centers must be virtualized to serve as cloud providers.

→ Nimbus, Eucalyptus, OpenNebula: open source software available to general public.

→ vSphere 4 – a proprietary OS for cloud resource virtualization & management over data centers.

* Trust Management in Virtualized Data Centers:

→ A VM encapsulates the state of the guest OS running inside it.

→ Encapsulated machine state can be copied & shared over the network & removed like a normal file; which proposes a challenge to VM security.

→ VMM can provide secure isolation. Once a hacker successfully enters the VMM (or) management VM, the whole system is in danger.

* Virtualization of CPU, Memory & I/O Devices:

→ To support virtualization, processors employ a special running mode & instructions.

** Hardware Support for virtualization:

→ All processors have atleast 2 modes; user mode and supervisor mode, to ensure controlled access of critical hardware.

→ Instructions running in supervisor mode are called privileged instructions.

→ XEN is a hypervisor for use in IA-32, x86-64, Itanium, PowerPC 970 hosts.

→ KVM (Linux Kernel virtualization) can support hardware-assisted virtualization and paravirtualization by using the Intel VT-x (or) AMD-v and VirtIO framework.

** CPU virtualization:

→ A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode.

Unprivileged instructions of VMs run directly on the host machine for higher efficiency.

→ Critical instructions are divided into 3 categories:

(1) Privileged Instructions: executed in privileged mode and will be trapped if executed outside this mode.

(2) Control-Sensitive Instructions: change the configuration of resources used.

(3) Behaviour Sensitive Instructions: have different behaviours depending on the configuration of resources.

→ A CPU architecture is virtualizable if it supports the ability to run the VM's privileged & unprivileged instructions in the CPU's user mode; while VMM runs in supervisor mode.

Ex: RISC CPUs ~~do~~ support virtualization.

* Hardware assisted CPU virtualization:

→ INTEL & AMD add an additional mode called privilege mode level (RING-1) to x86 processors.

∴ OS runs at Ring0 and hypervisor runs at Ring-1.

→ This removes the difficulty of implementing binary translation of full virtualization.

* Memory virtualization:

→ OS maintains mappings of virtual memory to machine memory using page tables.

→ Virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to physical memory of the VMs.

→ 2 stage mapping:
  * virtual memory to physical memory
  * physical memory to machine memory.

→ VMM is responsible for mapping the guest physical memory to the actual machine memory.

* I/O Virtualization:

↳ involves managing the routing of I/O requests b/w virtual devices & shared physical hardware.

→ 3 ways to implement

① Full device emulation:

→ All fns of a device or bus infrastructure such as device enumeration, identification, interrupts & DMA are replicated in software.

→ Software is located in VMM & acts as a virtual device.

→ Virtualization layer:
  • emulates the virtual device
  • remaps guest & real I/O addresses
  • multiplexes & drives the physical device.

② Para-Virtualization method:

→ used in Xen.

→ also known as split driver model consisting of a frontend driver & backend driver.

runs on domainU. runs on domain 0

→ interact using block of shared memory

→ Frontend driver: manages I/O requests.

→ Backend driver: responsible for managing the real I/O devices & multiplexing the I/O data.

③ Direct I/O virtualization:

→ focuses on networking for mainframes.

→ software based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical.

* Virtualization in Multi-core processors:

→ 2 main difficulties:

*→ Application programs must be parallelized to use all cores fully.

*→ Software must explicitly assign tasks to the cores, which is a very complex problem.

→ A multicore virtualization method was proposed to allow hardware designers to get an abstraction of low-level details of the processor cores.

* Virtualization Structures | Tools & Mechanisms:

→ Depending on position of virtualization layer

① Hypervisor and Xen Architecture:

→ hypervisor software sits directly b/w physical hardware & its OS.

→ Depending on functionality:

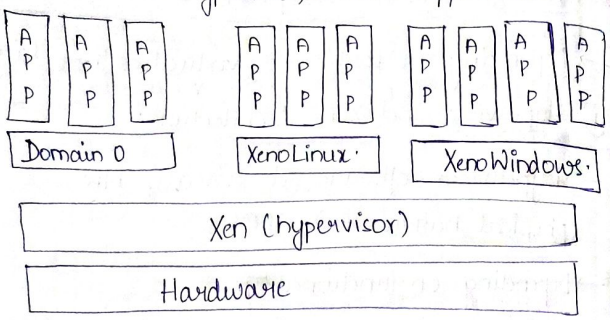
① Micro-kernel architecture (MS Hyper-V)
- ↳ includes only basic & unchanging fns.
- ↳ device drivers & other changeable components are outside the hypervisor.

② Monolithic hypervisor architecture (VMware ESX)
- ↳ implements all the mentioned fns, including those of the device drivers.

① is smaller than ②.

* Xen Architecture:
- ↳ microkernel hypervisor; implements all mechanisms, leaving policy to domain 0.
- ↳ does not include any device drivers
- ↳ size is very small.
- ↳ provides virtual environment located b/w hardware & OS.

→ Core components: hypervisor, kernel & applications.



→ guest OS has control ability = Domain 0.
→ Others : Domain U. ⇓ privileged guest OS of Xen.
→ Responsibilities of Domain 0 is to allocate & map hardware resources for the guest domains.

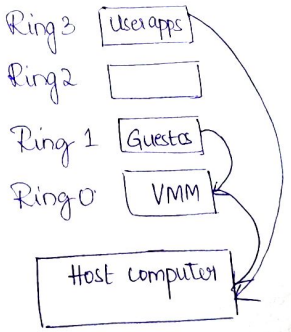
* Binary Translation with Full Virtualization:

→ Based on implementation, hardware virtualization is divided into:

① Full virtualization:
→ does not need to modify host OS.
→ relies on binary translation to trap & to virtualize the execution of certain sensitive, non-virtualizable instructions.
→ Binary translations incur a large performance overhead; hence critical instructions are only trapped into VMM.
→ Running non-critical instructions on hardware not only can promote efficiency, but also can ensure system security.

* Binary Translation of Guest OS Requests using a VMM:

→ VMM scans the instruction stream & identifies the privileged, control & behaviour sensitive instructions.
→ These instructions are trapped into VMM, which emulates the behaviour of these instructions. The method used is called binary translation.
→ The performance of full virtualization may not be ideal, because it involves binary translation which is rather time-consuming.



② Host Based virtualization:

→ alternative VM architecture, to install a virtualization layer on top of host OS.

→ guest OSes are installed & run on top of virtualization layer.

→ Advantages:

① User can install this VM architecture without modifying the host OS.

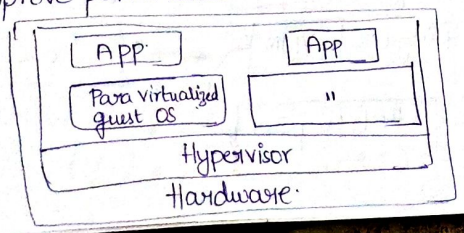
② Host based approach appeals to many host machine configurations.

* Para virtualization with Compiler Support:

→ Needs to modify guest OS.

→ Virtualization layer can be inserted at different positions in a machine software stack.

→ Attempts to reduce virtualization overhead & thus improve performance.

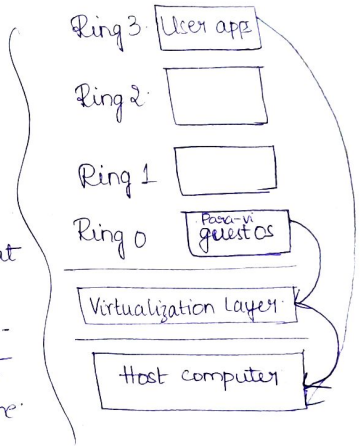


- Guest OSes are assisted by an intelligent compiler to replace the non-virtualizable OS instructions by hypercalls.

→ OS is responsible for managing the hardware & privileged instructions at Ring 0 & user applications at Ring 3.

→ Ex: KVM

- Unlike full, which intercepts & emulates privileged & sensitive instructions at run-time, para virtn. handles at compile time.



* Drawbacks:

(*) compatibility and portability

(*) Cost of maintaining para-virtualized OSes is high

→ Compared to full, para virtualization is relatively easy & more practical.

* Virtual Clusters & Resource Management:

→ Physical cluster is a collection of servers interconnected by a physical network such as LAN.

→ Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters.

* → Live VM migration steps & performance effects:

→ When a VM fails, its role could be replaced by another VM on a diff. node, as long as both run with same guest OS.

→ Ways to manage a virtual cluster:

1. Use a guest-based manager
2. Build a cluster manager on host systems.
3. Use an independent cluster manager on both the host and guest systems.
4. Use an integrated cluster on the guest & host systems.

→ VM can be in

1. inactive state: VM not enabled.
2. active state: perform some task.
3. paused state: waiting

4. suspended state: machine file & virtual resources are stored back to disk.

Slide 13

Stage 0: Pre-Migration: active VM on host A

Stage 1: Reservation: Container initialization

Stage: 2: Iterative pre-copy: Shadow paging

Stage: 3 Stop & Copy: Suspend VM on host A
Redirect traffic to host B.

Stage: 4 Commitment: VM state on host A released.

Stage 5: Activation: VM starts on Host B.

Theory

* Migration on Memory, Files & Network.

→ Issues:

① Memory Migration → moving memory instance from one host to another.

② File System "

③ Network " → enable remote systems to locate & communicate

④ Live migration of VM using XEN.

Slide 22

↳ Internet Suspend Resume (ISR) exploits temporal locality as memory states.

② Simple way to achieve this is to provide each VM with its own virtual disk which file system is mapped.