Cloud computing - Virtualization Technology

Date @November 18, 2024

Basic definitions

Virtualization

Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer (processors, memory, storage) to be divided into multiple virtual computers, commonly called Virtual Machines. Each VM runs its own operating system and behaves line an independent computer, even though it is running on just a portion of the actual underlying computer hardware.

Virtualization is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. It enables cloud providers to serve users with their existing physical computer hardware; it enables cloud users to purchase only the computing resources they need when they need it and to scale those resources cost-effectively as their workloads grow.

Virtual machines

Virtual machines (VMs) are virtual environments that simulate a physical computer (with its OS) in software form. They normally comprise several files containing the VM's configuration, the storage for the virtual hard drive and some snapshots of the VM that preserve its state at a particular point in time.

Advantages of VMs

- Resource utilization and improved ROI (return in investment): since multiple VMs run on a single physical computer, customers do not have to buy a new server every time they want to run another OS, and they can get more return from each piece of hardware they already own
- Flexibility: creating a VM is faster and easier than installing an OS on a physical server because you can clone a VM with the OS already installed. Developers and software testers can create new environments on demand to handle new tasks as they arise

- Scalability: with cloud computing, it is easier to deploy multiple copies of the same virtual machine to better serve increases in load
- Portability: VMs can be relocated as needed among the physical computers in a network; this makes it possible to allocate workloads to servers that have spare computing power. VMs can even move between on-permission and cloud environments, making them useful for hybrid cloud scenarios in which you share computing resources between your data center and a cloud service provider
- security

Host OS & Guest OS

An host OS is an OS running on the physical machine hosting the VMs. It may include the Hypervisor & Virtual Machine Monitor, or be replaced by them for the hardware virtualization.

A guest OS is an OS running in the VM, which should not be aware of running in a virtualized environment.

Hypervisor & Virtual Machine Monitor

The VMM is responsible for virtualizing system resources for the guest, including virtual devices and guest memory allocation. An Hypervisor, also known as virtualization platform, is a software responsible for direct control over VMs, including CPU state management guest enter/exit, memory protection.

These two terms are usually not strictly distinguished, and Hypervisor is commonly translated as a VMM. For simplicity, we use the term Hypervisor to indicate both in these notes.

What are hypervisors

A hypervisor is the software layer that coordinates VMs. It serves as an interface between the VM and the underlying physical hardware, ensuring that each has access to the physical resources it needs to execute. It also ensures that the VMs do not interfere with each other by impinging on each other’s memory space or compute cycles.

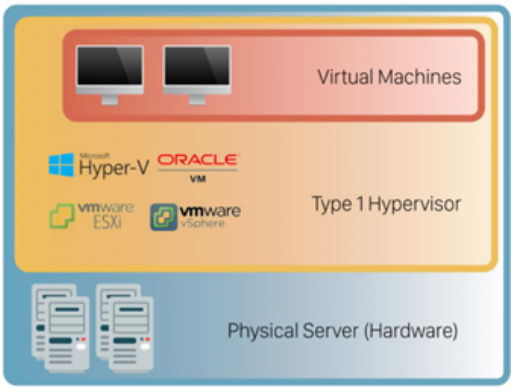
Characteristics of hypervisors

- performance: ideally, hypervisors should support guest OS performance close to native speed
- management tools: running VMs is not the only thing you must manage when using a hypervisor. You must provision the VMs, maintain them, audit them, and clean up disused ones to prevent VM sprawl- The hypervisor architecture must be comprehensive of management tools.

- ecosystem: third-party developers that can support the hypervisor with their own agents and plugins that offer capabilities, such as backup and fail-over management
- live migration: this enables to move VMs between hypervisors on a different physical machines without stopping them, which can be useful for both fail-over and workload balancing
- cost: the cost of the hypervisor itself is not the only expense; the management software that makes it scalable to support an enterprise environment can often be expensive

Type 1 Hypervisors

It runs directly on the underlying computer's physical hardware, interacting directly with its CPU, memory, and physical storage. For this reason, Type 1 hypervisors are also referred to as bare-metal hypervisors. A type 1 hypervisor takes the place of the host operating system.



The advantages are:

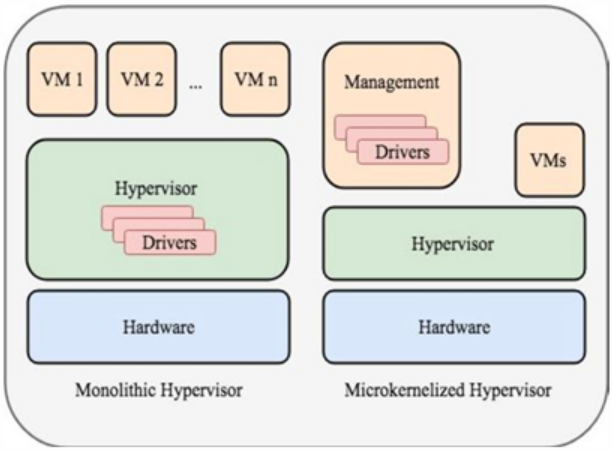
- performance: is highly efficient because it has direct access to physical hardware
- security: has direct access to hardware without an additional OS layer; this direct connection decreases the attack surface
- VM mobility: enables moving VMs between physical servers automatically and easily; in case of a hardware failure, the detection and restoration procedure takes place automatically and seamlessly
- resource over-allocation: more resources than the available ones can be assigned to VMs; if you have 128GB of RAM on your server and eight virtual machines, you can assign 24GB of RAM to each. This totals 192GB of RAM, but VMs themselves will not consume all 24GB from the physical server. The VMs detect they have 24GB when they only use the amount of RAM they need to perform particular tasks.

The disadvantages are:

- complicated management: to create virtual instances, you need a management console set up on another machine; using the console, you can connect to the hypervisor on the server and manage your virtual environment
- price: depending on what functionalities you need, the license cost for management consoles varies substantially

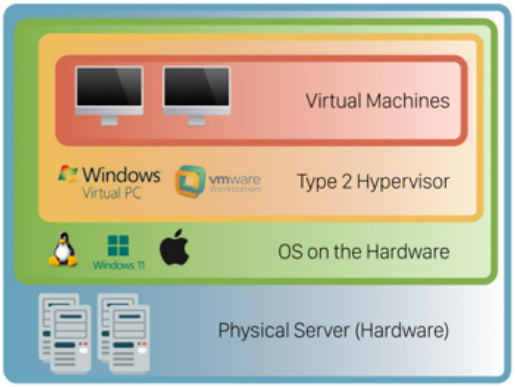
Type 1 hypervisors can be classified in:

- monolithic: the hypervisor is a single thick layer that also includes most of the required components, such as the kernel, device drivers and the I/O stack
- microkernelized: the hypervisor is a very thin, specialized one that only performs the core tasks of ensuring partition isolation and memory management; this layer does not include the I/O stack or device drivers. Virtualization stack and hw-specific device drivers are located in a specialized partition called the parent partition



Type 2 Hypervisor

This type of hypervisor runs as an application in an OS and rarely show up in server-based environments. Instead, they are suitable for individual PC users needing to run multiple operating systems. For this reasons it is also referred to as residence/hosted hypervisor. It often requires additional toolkits for users to install into the guest OS. These tools provide enhanced connections between the guest and the host OS, often enabling the user to cut and paste between the two or access host OS files and folders from within the guest VM.



The advantages are:

- easy to manage: enables quick and easy access to an alternative guest OS alongside the primary one running on the host system; this makes it great for end-user productivity and a consumer might use it to access their Linux development tools while using a speech dictation system only found in windows, for example
- allows access to additional productivity tools: users can use the tools available on other operating systems alongside their primary OS
- cost: it is less expensive

The disadvantages are:

- performance: as it must access computing, memory and network resources via the host OS, it introduces latency
- less flexible resource management: bare-metal hypervisors can dynamically allocate available resources depending on the current needs of a particular VM. Furthermore, occupies whatever the user allocates to a virtual machine, for example, when a user assigns 8GB of RAM to a VM, that amount will be taken up even if the VM is using only a fraction of it. If the host machine has 32GB of RAM and the user creates three VMs with 8GB each, they are left with 8GB of RAM to keep the physical machine running. Creating another VM with 8GB of RAM would bring down the system.
- security: it suns on top of an OS, introducing a potential vulnerability since attackers may use potential vulnerabilities of the OS to gain access to virtual machines.

Virtualization techniques and types

Virtual hardware

The virtual hardware profile defines the characteristics of the hardware provided to the guest OS. The duty of the hypervisor is to emulate the virtual hardware, with the exact

characteristics specified in the above hardware profile. The real hardware may have a little to do with the virtualized hardware.

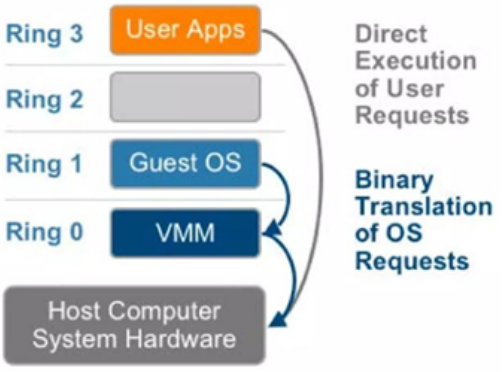
Virtualization techniques

There are multiple approaches to run the Guest OS:

- full virtualization
- paravirtualization (also known as OS-assisted virtualization)
- HW-assisted virtualization (also known as hardware virtual machine HVM)

Full virtualization

In full virtualization the guest OS is completely unaware that it is a guest managed by a hypervisor. Non-critical instructions run on the hardware directly while critical instructions are discovered and replaced by the hypervisor with a series of non-critical instructions in order to emulate what should happen with the critical ones. The hypervisor binary translates all OS calls on the fly.



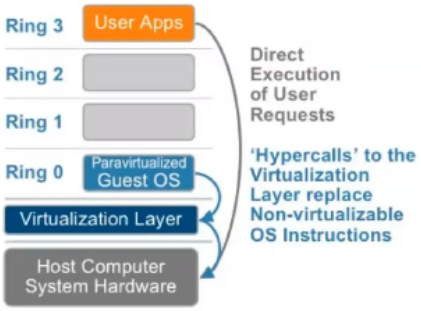
Example: handling CLI

When a guest O tries to execute CLI, the hypervisor intercepts it and replaces it with safe, virtualized code that achieves the desired effect disrupting the host. Instead of letting the guest execute CLI on the real hardware, the hypervisor rewrites it into code that only affects the guest's virtual CPU (vCPU) state, not the real CPU. The real CPU's interrupt state remains unaffected, so other virtual machines and the host continue to handle interrupts as normal.

Paravirtualization

The guest OS is this case is aware that is a guest managed by a hypervisor. The hypervisor provides hypercall APIs for critical kernel operations such as memory management, interrupt handling and time keeping. The guest OS is modified to replace sensitive instructions with hypercalls that communicate directly with the hypervisor.

While it is very difficult to build the more binary translation support necessary for full virtualization, modifying the Guest OS to enable paravirtualization is relatively easy.



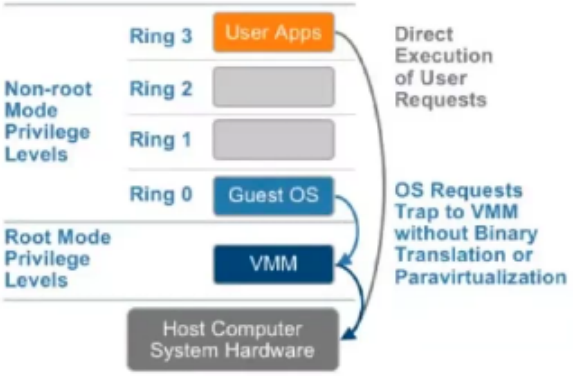
Example: handling CLI

The guest OS has been re-engineered to avoid sensitive instruction altogether, using hypercalls as safe substitutes. The guest does not execute CLI directly but it makes a hypercall to the hypervisor to request this operation. The hypervisor sets the interrupt flag in the vCPU state for the guest, “disabling interrupts” from the guest’s perspective. The hypervisor can still maintain control of the physical CPU interrupt state, keeping the host system stable and other VMs unaffected.

HW-assisted virtualization

The guest OS is not aware to be a guest. The CPU provides support for virtualization allowing the hypervisor to handle sensitive instructions efficiently by relying on hardware traps rather than software emulation. Hardware trap is a mechanism in hardware which triggers the execution of some code in the OS.

The guest OS does not need modifications here and the hypervisor does not have to expose APIs. This improves efficiency but requires the host system to have an expensive processor that supports the technology.



Example: handling CLI

The guest OS in this case executes the CLI. The CPU notices that the CLI instruction is being executed in the guest OS and triggers a VM-Exit. This is a hardware-assisted trap

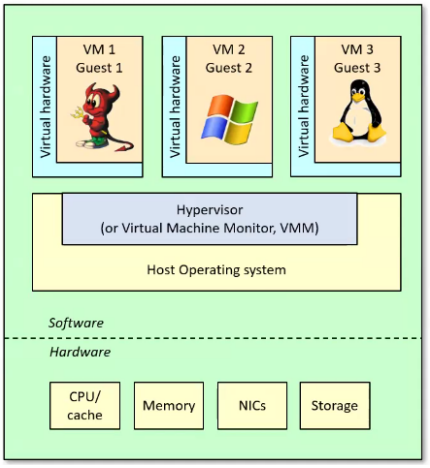
that causes the CPU to switch from running in the guest's context to the hypervisor's context (hypervisor mode). At this point, the hypervisor takes over and performs the necessary steps to safely handle the instruction.

Types of virtualization

Virtualization types are about what we want to virtualize: a computer, a network, a storage, etc.

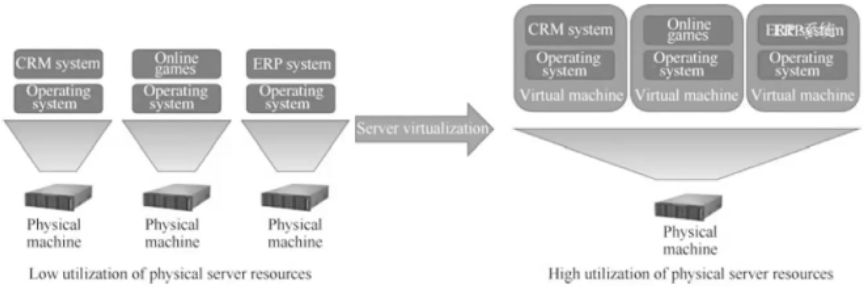
System virtualization

This is the most used virtualization type. It realizes the separation of operating system and the physical machine, so that one or more virtual operating systems can be installed and run on a physical machine at the same time.



From the perspective of the applications inside the OS, there is no difference between the virtual OS and the OS directly installed on the physical machine. In system virtualization, multiple OSs can run simultaneously on the same physical machine without affecting each other, reusing physical machine resources.

Server (an OS serving requests) virtualization applies system virtualization technology to servers, virtualizing one server into several servers



Desktop (our personal computer which sends requests to the server) virtualization lets you run multiple desktop operating systems, each in its own VM on the same computer. There are four types of desktop virtualization:

- local desktop virtualization
- virtual desktop infrastructure (VDI)
- remote desktop virtualization
- desktop-as-a-service (DaaS)

Local desktop virtualization

It runs hypervisor on a local computer, enabling the user to run one or more additional OSs on that computer and switch from one OS to another as needed without changing anything about the primary OS. This desktop virtualization model is directly managed by the end user.

Virtual desktop infrastructure

It hosts desktop environments on a centralized server and deploys them to end-users on request. With VDI, a hypervisor segments servers into virtual machines (VMs) that host these virtual desktops. All processing is done on the host server. The virtual desktop image is delivered over a network to an endpoint device which allows the user to interact with the OS and its applications as if they were running locally. The endpoint may be a traditional PC, thin client device or a mobile device. In a VDI environment each user can access their own centrally hosted VM, access a shared VM, and IT can have CPU, memory and disk capacity reserved to a specific user based on his specific computing needs.

Remote desktop virtualization

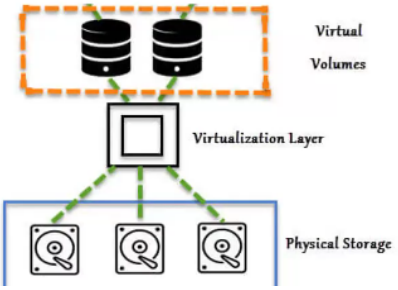
Microsoft RDS (remote desktop service) provides a centralized platform for delivering and managing Windows-based applications and desktops to user via Microsoft's remote desktop protocol (RDP). RDS is a feature of Microsoft Windows Server that lets users connect to a shared desktop environment or individual applications running on a remote server. A single server can support multiple active user sessions, which can be customized on a per user basis, but server resources are not dedicated to a particular user. (You don't have remote VMs; you have them in VDI)

Desktop-as-a-service

This is a way of delivering complete virtual desktop environments to users including OSs, applications, files and user preferences from the cloud. The desktops run in virtual machines hosted on compute, storage and network infrastructure managed by the cloud provider. Users can access their desktop environment from a wide variety of devices, including PCs, laptops, tablets and some smartphones.

Storage virtualization

Storage virtualization enables all the storage devices on the network to be accessed and managed as a single storage device. Storage virtualization masses all blocks of storage into a single shared pool from which they can be assigned to any VM on the network as needed. Storage virtualization makes it easier to provision storage for VMs and makes maximum use of all available storage on the network.



Application virtualization

Application virtualization runs applications software without installing it directly on the user's OS. This differs from complete desktop virtualization because only the application runs in a virtual environment, while the OS on the end user's device runs as usual.

There are three types of application virtualization:

- local application virtualization: the entire application runs on the endpoint device but runs in a runtime environment instead of on the native hardware
- application streaming: the application lives on a server which sends small components of the software to run on the end user's device when needed
- server-based application virtualization: the application runs entirely on a server that sens only its user interface to the client device

Popular hypervisors

VMware

VMware offers Type 1 and Type 2 hypervisors. ESXi and VSphere for Type 1 and VMware fusion, Workstation and VirtualBox for Type 2.

The main characteristics of VMware ESX and ESXi are enterpirse virtualization platforms offered by VMware, ESX runs on bare metal, without an OS, and have a monolithic architecture. ESX includes its own Linux kernel, which is started first and then loads VMware's VMkernel component, while ESXi does not contain the Linux kernel. It directly loads from VMkernel.

Workstation is for Windows and Linux host OSs, while VMware fusion is for MacOs host OSs. Virtualbox is instead on all the three OSs.

Hyper-V

Microsoft has his own Type 1 hypervisor, that is Hyper-V hypervisor. It has a microkernelized design and it runs a lightweight version of Windows. It comes in two variants: hyper-V server, a stand-alone version including full hyper-V functionality, and an installable role version for Windows server.

KVM

Kernel-based virtual machine comes designed as small light kernel module to leverage the facilities provided by hardware support of virtualization. It is almost always defined as a Type 1 hypervisor but there is a huge discussion on this because typically in Type 1 hypervisor we have the software replacing the host OS and in Type 2 hypervisors we have an application running on a host OS and then managing all the translations; in KVMs we have a set of drivers which are installed on the Linux kernel and for this, the hypervisor becomes Linux itself.