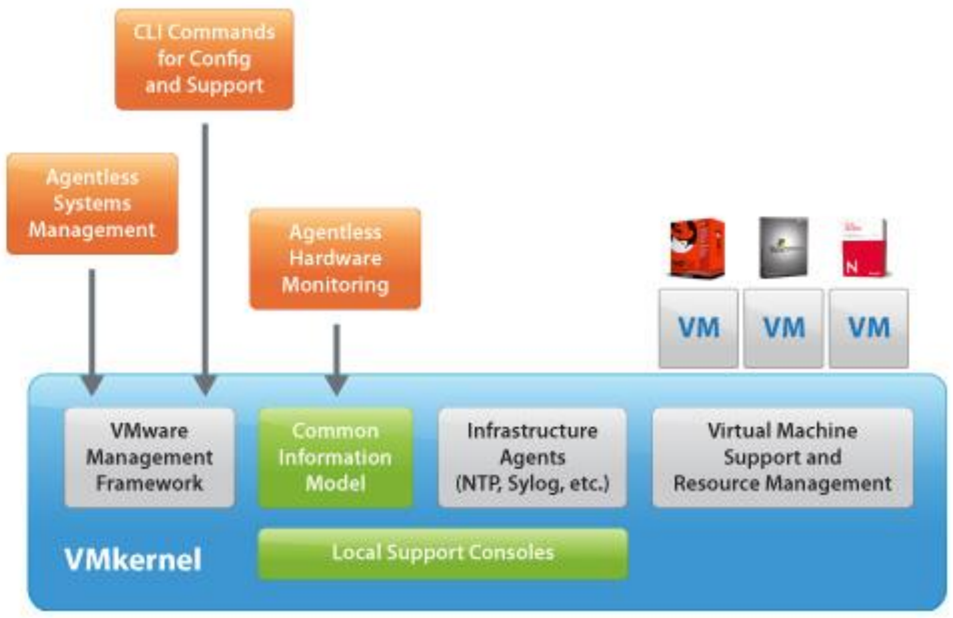
❖ VMware ESXi

ESX server is a virtualization platform, and a flagship enterprise by VMware. Usually, this is available in two versions – ESXi server and ESX server. Basically ESX server is an enterprise level virtualization tool. It makes use of different services which manage multiple virtual machines. Usually, these machines have greater efficiency and reliability, as compared to the basic Server products, offered by VMware. The reason why this server ensures top-notch performance is that it runs on bare metal. This means that the user can install the software of the ESX server directly to his computer. There is no need of an additional operating system that will make it run.

The basic system on which the ESX server has been designed is controlled by VMkernel. This is a microkernel based on the Linux kernel and it saves the valuable resources. This eliminates the need of running an underlying operating system just beneath the virtual machines. After the setup is complete, the resources are divided from the hardcore physical hardware and generates numerous copies of the virtual hardware so that the virtual machines can use those.

VMkernel ensures all the processes can be run smoothly on the system itself, including the virtual machines, agents and management applications. In fact, it can control all the hardware devices, based on the server and it can also manage the resources for the applications. The main processes that run on VMkernel, include –

- **Virtual machine monitor –** This ensures the process of providing the execution environment for a virtual machine. It also initiates a process called VMX. Each virtual machine that is in running condition has its own VMX as well as VMM process.
- **Direct Console User Interface or DCUI** – This is a low level management and configuration interface. This component is accessible right through the console of the server. However, this component is primarily used for some basic level configuration that are usually done in the initial phase.
- **CIM system** – The Common Information Model or CIM is an interface which enables one to control the hardware-level management even from the remote applications through a set of standardized APIs.
- **Various agents** – These components are usually used for enabling one to control the high-end VMware infrastructure even from remote applications.


CLI Commands for Config and Support
Agentless Systems Management
Agentless Hardware Monitoring
VM
VM
VM
VMware Management Framework
Common Information Model
Infrastructure Agents (NTP, Sylog, etc.)
Virtual Machine Support and Resource Management
VMkernel
Local Support Consoles

One of the greatest features of the ESX server is that it has the capacity to overcommit the capability to memorize. This refers to the fact that the total memory of the virtual machine can easily exceed the original physical memory capacity of the server. This ensures better utilization of memory in the servers. All these procedures are controlled by the service console. In fact, this console acts as the management software for the ESX server as well as its operating system.

ESX server architecture has been designed in an innovative way so that it can be operated from a general operating system that offers increased reliability, simplified management and better security. The architecture has been designed in a compact fashion so that it can be directly integrated to a server hardware that is virtually optimized. This is important since this enables smooth deployment, hassle free configuration and quick installation.

Basically, ESX server is a virtualization layer which abstracts the memory, networking resources, storage and the processor of a physical server into a number of virtual machines. These machines run in a parallel fashion, in an isolated and secured way. A particular copy of the ESX server runs on each physical x86 machine. Basically ESX server communicates with some of the other resources through the Host Agent. While doing this, the server makes use of the VMware infrastructure API or VI API. Using this specific infrastructure, both third party software as well as VMware can gather information from the VI API.

Who should use ESX server?

- The ESX server is basically designed for companies that require to streamline the hardware server to a great extent. This server is also used for performing instant deployment of new servers. In fact, this server is also used for maintaining the existing servers that are in use. This helps ensure zero downtime.
- One of the most interesting usages of ESX server is for disaster recovery. Since the ESX server is dependent on hardware, the operating systems and the applications that run on this server can be ported instantly. Since the hardware and the virtual specifications of this server are uniform in nature, this ensures smooth maintenance and deployment. This also makes the process of managing virtual machines from remote locations absolutely hassle free.

In order to run the essential processor instructors perfectly, ESX server needs some special CPU hardware from AMD or Intel. You will also need a consistent storage solution so that you can store all the information on the virtual machines as well as the virtual hard disks of those machines.

❖ XEN Architecture

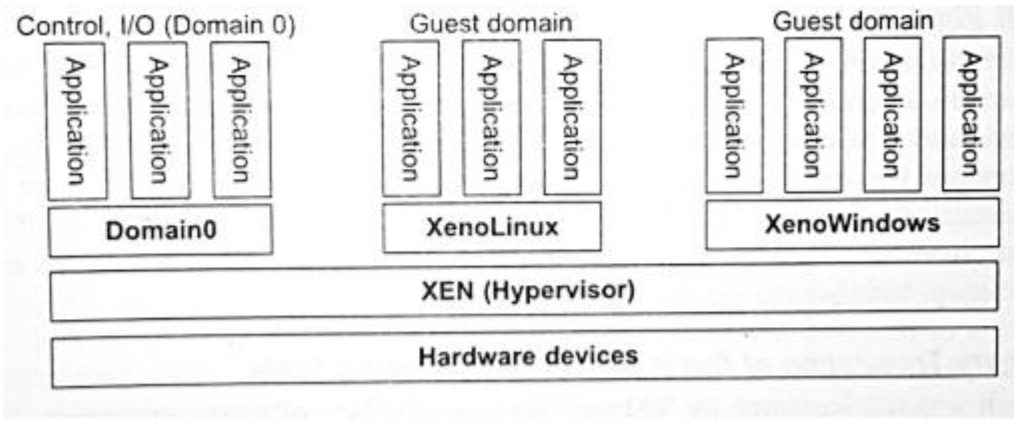
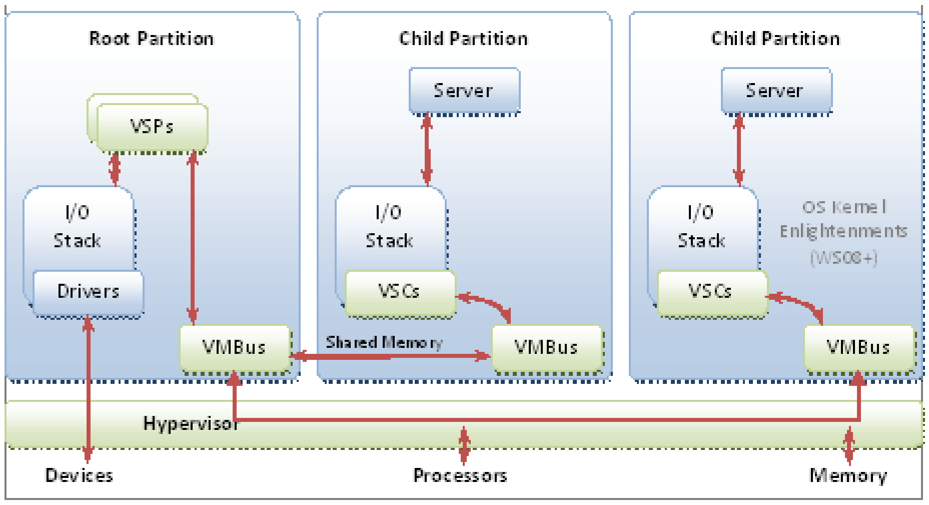


Fig. Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a microkernel hypervisor, which separates the policy from the mechanism.
- The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0, as shown in figure does not include any device drivers natively.It just provides a mechanism by which a guest OScan have direct access to the physical devices. As a result, the size of the Xen hypervisor is keptrather small.
- Xen provides a virtual environment located between the hardware and the OS.
- A number of vendors are in the process of developing commercial Xen hypervisors, among the mare Citrix XenServer and Oracle VM.
- The core components of a Xen system are the hypervisor, kernel, and applications.
- The organization of the three components is important.
- Like other virtualization systems, many guest OSescan run on top of the hypervisor.
- The guest OS, which has control ability, is called Domain 0, and the others are called Domain U. Domain 0 is a privileged guest OS of Xen.
- It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices. Therefore, one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains (the Domain U domains).
- For example, Xen is based on Linux and its security level is C2. Its management VM is named Domain 0, which has the privilege to manage other VMs implemented on the same host.
- If Domain0 is compromised, the hacker can control the entire system. So, in the VM system, security policies are needed to improve the security of Domain 0.

- Domain 0, behaving as a VMM, allows users to create, copy, save, read, modify, share, migrate, and roll back VMs as easily as manipulating a file, which flexibly provides tremendous benefits for users.
- It also brings a series of security problems during the software life cycle and data lifetime.
- Traditionally, a machine's lifetime can be envisioned as a straight line where the current state of the machine is a point that progresses monotonically as the software executes.
- During this time, configuration changes are made, software is installed, and patches are applied. In such an environment, the VM state is in to a tree: At any point, execution can go into N different branches where multiple instances of a VM can exist at any point in this tree at any given time. VMs are allowed to roll back to previous states in their execution (e.g., to fix configuration errors) or rerun from the same point many times (e.g., as a means of distributing dynamic content or circulating a "live" system image).

❖ Hyper – V Architecture



Hyper-V features a Type 1 hypervisor-based architecture. The hypervisor virtualizes processors and memory and provides mechanisms for the virtualization stack in the root partition to manage child partitions (virtual machines) and expose services such as I/O devices to the virtual machines.

The root partition owns and has direct access to the physical I/O devices. The virtualization stack in the root partition provides a memory manager for virtual machines, management APIs, and virtualized I/O devices. It also implements emulated devices such as the integrated device electronics (IDE) disk controller and PS/2 input device port, and it supports Hyper-V-specific synthetic devices for increased performance and reduced overhead.

The Hyper-V-specific I/O architecture consists of virtualization service providers (VSPs) in the root partition and virtualization service clients (VSCs) in the child partition. Each service is exposed as a device over VMBus, which acts as an I/O bus and enables high-performance communication between virtual machines that use mechanisms such as shared memory. The guest operating system's Plug and Play manager enumerates these devices, including VMBus, and loads the appropriate device drivers (virtual service clients). Services other than I/O are also exposed through this architecture.

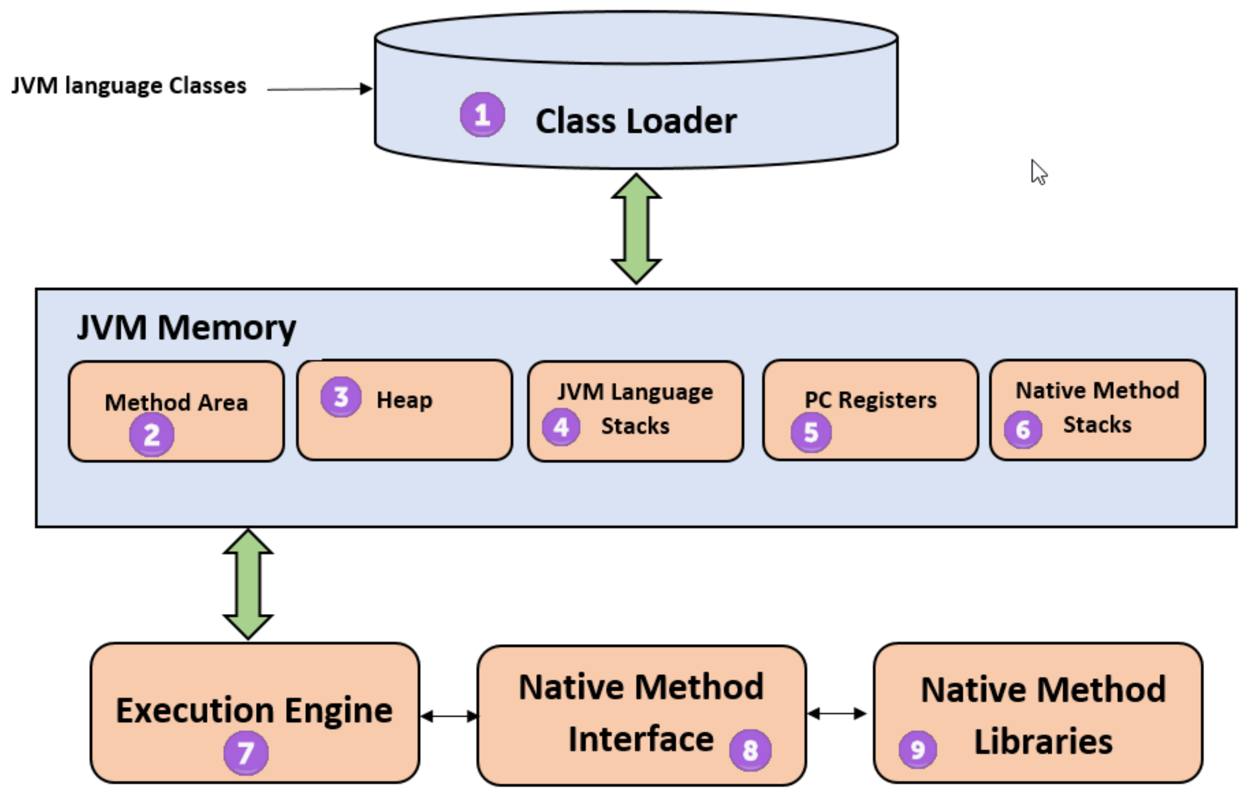
Starting with Windows Server 2008, the operating system features enlightenments to optimize its behavior when it is running in virtual machines. The benefits include reducing the cost of memory virtualization, improving multicore scalability, and decreasing the background CPU usage of the guest operating system.

❖ Java Virtual Machine Architecture

Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Run Environment (JRE). In other programming languages, the compiler produces machine code for a particular system. However, Java compiler produces code for a Virtual Machine known as Java Virtual Machine.

- **Here is how JVM works**
- First, Java code is complied into bytecode. This bytecode gets interpreted on different machines
- Between host system and Java source, Bytecode is an intermediary language.
- JVM is responsible for allocating memory space.

Architecture of JVM. It contains classloader, memory area, execution engine etc.



1) ClassLoader

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

2) Method Area

JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods.

3) Heap

All the Objects, their related instance variables, and arrays are stored in the heap. This memory is common and shared across multiple threads.

4) JVM language Stacks

Java language Stacks store local variables, and it's partial results. Each thread has its own JVM stack, created simultaneously as the thread is created. A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

5) PC Registers

PC register store the address of the Java virtual machine instruction which is currently executing. In Java, each thread has its separate PC register.

6) Native Method Stacks

Native method stacks hold the instruction of native code depends on the native library. It is written in another language instead of Java.

7) Execution Engine

It is a type of software used to test hardware, software, or complete systems. The test execution engine never carries any information about the tested product.

8) Native Method interface

The Native Method Interface is a programming framework. It allows Java code which is running in a JVM to call by libraries and native applications.

9) Native Method Libraries

Native Libraries is a collection of the Native Libraries(C, C++) which are needed by the Execution Engine.