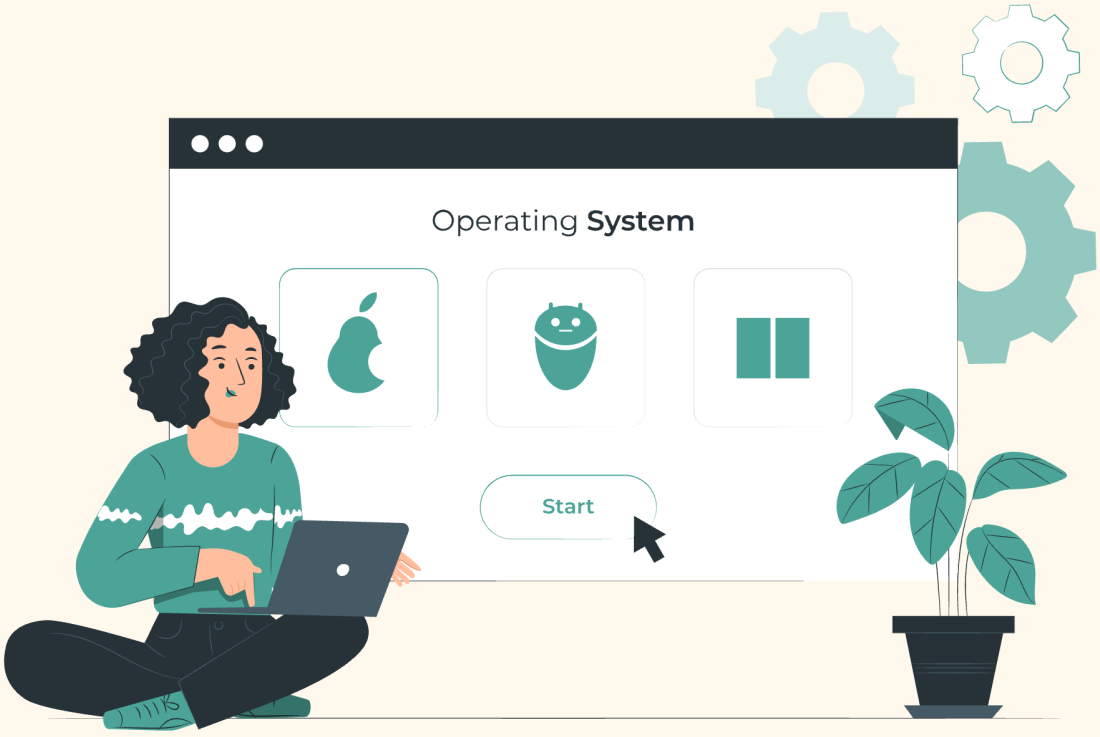
Operating System
PCC-CS 502
Report for CA 2

VIRTUAL MACHINE

—

Badhan Biswas

CSE - 11000120019



OVERVIEW

A Virtual Machine, commonly shortened to just VM, is no different than any other physical computer like a laptop, smart phone or server. It has a CPU, memory, disks to store your files and can connect to the internet if needed. While the parts that make up your computer (called hardware) are physical and tangible, VMs are often thought of as virtual computers or software-defined computers within physical servers, existing only as code.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.
Virtual machines (VMs) allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

In this short report we will know about virtual machines in detail and its use cases.

Table of

CONTENTS

Virtual Machine

A virtual machine is a virtual representation, or emulation, of a physical computer. They are often referred to as a guest while the physical machine they run on is referred to as the host.

Virtualization makes it possible to create multiple virtual machines, each with their own operating system (OS) and applications, on a single physical machine. A VM cannot interact directly with a physical computer. Instead, it needs a lightweight software layer called a hypervisor to coordinate between it and the underlying physical hardware. The hypervisor allocates physical computing resources—such as processors, memory, and storage—to each VM. It keeps each VM separate from others so they don't interfere with each other.

While this technology can go by many names, including virtual server, virtual server instance (VSI) and virtual private server (VPS), this article will simply refer to them as virtual machines.

Types of VM

Users can choose from two different types of virtual machines—process VMs and system VMs:

A ***process virtual machine*** allows a single process to run as an application on a host machine, providing a platform-independent programming environment by masking the information of the underlying hardware or operating system. An example of a process VM is the Java Virtual Machine, which enables any operating system to run Java applications as if they were native to that system.

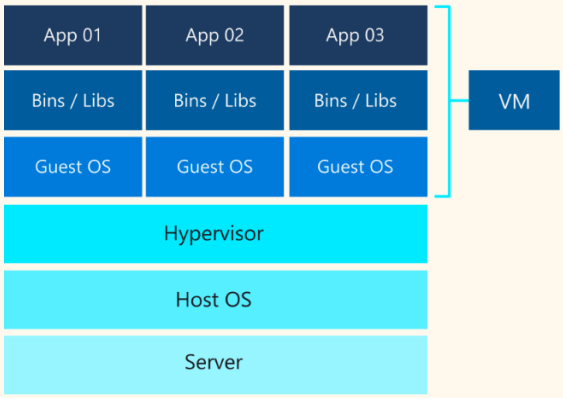
A ***system virtual machine*** is fully virtualized to substitute for a physical machine. A system platform supports the sharing of a host computer's physical resources between multiple virtual machines, each running its own copy of the operating system. This virtualization process relies on a hypervisor, which can run on bare hardware, such as VMware ESXi, or on top of an operating system.

How virtualization works

When a hypervisor is used on a physical computer or server, (also known as bare metal server), it allows the physical computer to separate its operating system and applications from its hardware. Then, it can divide itself into several independent "virtual machines."

Each of these new virtual machines can then run their own operating systems and applications independently while still sharing the original resources from the bare metal server, which the hypervisor manages. Those resources include memory, RAM, storage, etc.

What are hypervisors?



Before hypervisors hit the mainstream, most physical computers could only run one operating system (OS) at a time. This made them stable because the computing hardware only had to handle requests from that one OS. The downside of this approach was that it wasted resources because the operating system couldn't always use all of the computer's power.

A hypervisor solves that problem. It is a small software layer that enables multiple operating systems to run alongside each other, sharing the same physical computing resources. These operating systems come as virtual machines (VMs)—files that mimic an entire computing hardware environment in software.

The hypervisor, also known as a virtual machine monitor (VMM), manages these VMs as they run alongside each other. It separates VMs from each other logically, assigning each its own slice of the underlying computing power, memory, and storage. This prevents the VMs from interfering with each other; so if, for example, one OS suffers a crash or a security compromise, the others survive.

Characteristics of Hypervisors

There are different categories of hypervisors and different brands of hypervisors within each category. The market has matured to make hypervisors a commodity product in the enterprise space, but there are still differentiating factors that should guide your choice. Here's what to look for:

- ***Performance***: Look for benchmark data that show how well the hypervisor performs in a production environment. Ideally, bare-metal hypervisors should support guest OS performance close to native speeds.

- ***Ecosystem***: You will need good documentation and technical support to implement and manage hypervisors across multiple physical servers at scale. Also, look for a healthy community of third-party developers that can support the hypervisor with their own agents and plugins that offer capabilities, such as backup and restore capacity analysis and fail-over management.

- ***Management tools***: Running VMs isn't 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." Ensure that the vendor or third-party community supports the hypervisor architecture with comprehensive management tools.

- ***Live migration***: This enables you to move VMs between hypervisors on different physical machines without stopping them, which can be useful for both fail-over and workload balancing.

- ***Cost***: Consider the cost and fee structure involved in licensing hypervisor technology. Don't just think about the cost of the hypervisor itself. The management software that makes it scalable to support an enterprise environment can often be expensive. Lastly, examine the vendor's licensing structure, which may change depending on whether you deploy it in the cloud or locally.

Type 1 vs. Type 2

There are two broad categories of hypervisors: Type 1 and Type 2.

Type 1 hypervisor

A Type 1 hypervisor 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.

- ***Pros:*** Type 1 hypervisors are highly efficient because they have direct access to physical hardware. This also increases their security, because there is nothing in between them and the CPU that an attacker could compromise.

- ***Cons:*** A Type 1 hypervisor often needs a separate management machine to administer different VMs and control the host hardware.

Type 2 hypervisor

A Type 2 hypervisor doesn't run directly on the underlying hardware. Instead, it runs as an application in an OS. Type 2 hypervisors rarely show up in server-based environments. Instead, they're suitable for individual PC users needing to run multiple operating systems. Examples include engineers, security professionals analyzing malware, and business users that need access to applications only available on other software platforms.

Type 2 hypervisors often feature 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.

- ***Pros:*** A Type 2 hypervisor 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. A consumer might use it to access their favorite Linux-based development tools while using a speech dictation system only found in Windows, for example.
- ***Cons***: A Type 2 hypervisor must access computing, memory, and network resources via the host OS, which has primary access to the physical machine. This introduces latency issues, affecting performance. It also introduces potential security risks if an attacker compromises the host OS because they could then manipulate any guest OS running in the Type 2 hypervisor.

Examples of Hypervisors

VMware hypervisors

- ***ESXi hypervisor***: VMware ESXi (Elastic Sky X Integrated) is a **Type 1** (or bare-metal) hypervisor targeting server virtualization in the data center. ESXi manages collections of VMware virtual machines.
- ***VMware Fusion***: VMware also offers **Type 2** hypervisor products for desktop and laptop users.This is the company's MacOS-focused offering, which lets Mac users run a large range of guest operating systems.

Use cases for VMs

VMs have several uses, both for enterprise IT administrators and users. Here are a few options:

- ***Cloud computing***: For the last 10+ years, VMs have been the fundamental unit of compute in the cloud, enabling dozens of different types of applications and workloads to run and scale successfully.
- ***Support DevOps***: VMs are a great way to support enterprise developers, who can configure VM templates with the settings for their software development and testing processes. They can create VMs for specific tasks such as static software tests, including these steps in an automated development workflow. This all helps streamline the DevOps toolchain.
- ***Test a new operating system***: A VM lets you test-drive a new operating system on your desktop without affecting your primary OS.
- ***Investigate malware***: VMs are useful for malware researchers that frequently need fresh machines on which to test malicious programs.
- ***Run incompatible software***: Some users may prefer one OS while still needing a program that is only available in another. One good example is the Dragon range of voice dictation software. Its vendor, Nuance, has discontinued the macOS version of its product. However, running a desktop-focused hypervisor—such as VMware Fusion or Parallels—enables you to run Windows in a VM, giving you access to that version of the software.
- ***Browse securely***: Using a virtual machine for browsing enables you to visit sites without worrying about infection. You can take a snapshot of your machine and then roll back to it after each browsing session. This is something that a user could set up themselves, using a Type 2 desktop hypervisor. Alternatively, an admin could provide a temporary virtual desktop located on the server.

What are the benefits of using VMs?

While virtual machines run like individual computers with individual operating systems and applications, they have the advantage of remaining completely independent of one another and the physical host machine. A piece of software called a hypervisor or virtual machine manager, lets you run different operating systems on different virtual machines at the same time. This makes it possible to run Linux VMs, for example, on a Windows OS or to run an earlier version of Windows on a more current Windows OS.

And, because VMs are independent of each other, they are also extremely portable. You can move a VM on a hypervisor to another hypervisor on a completely different machine almost instantaneously.

Because of their flexibility and portability, virtual machines provide many benefits, such as:

- ***Cost savings***—running multiple virtual environments from one piece of infrastructure means that you can drastically reduce your physical infrastructure footprint. This boosts your bottom line—decreasing the need to maintain nearly as many servers and saving on maintenance costs and electricity.
- ***Agility and speed***—Spinning up a VM is relatively easy and quick and is much simpler than provisioning an entire new environment for your developers. Virtualisation makes the process of running dev-test scenarios a lot quicker.
- ***Lowered downtime***—VMs are so portable and easy to move from one hypervisor to another on a different machine—this means that they are a great solution for backup, in the event the host goes down unexpectedly.
- ***Scalability***—VMs allow you to more easily scale your apps by adding more physical or virtual servers to distribute the workload across multiple VMs. As a result you can increase the availability and performance of your apps.
- ***Security benefits***— Because virtual machines run in multiple operating systems, using a guest operating system on a VM allows you to run apps of questionable security and protects your host operating system. VMs also allow for better security forensics and are often used to safely study computer viruses, isolating the viruses to avoid risking their host computer.

Disadvantages of Virtual Machine

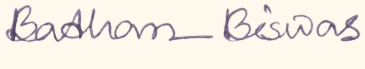
- ***Cos***t: Running a virtual machine with a cloud computing service is generally considered to be expensive. The upfront costs involved in a virtual machine makes it costly. And depending on the need this cost varies. If there is more need, more investments need to be done.
- ***Performance***: Even though the machines are virtualized in a virtual machine, it still relies on the resources from the host machine. A computer needs to be powerful enough to run several virtual machines on a single host computer. If its power is not sufficient enough, it will cause stability issues in the performance.
- ***Efficiency***: In terms of hardware accessibility a virtual machine is less efficient. It cannot access the hardware directly. And also its speed is not sufficient for most IT firms. This makes them use a system that is balanced between virtual and physical.
- ***Complexity***: A virtual machine is a complex system. What makes it complex is the multiple local area networks it is equipped with. Therefore, in case of any failure it will be difficult to figure out where the fault has occurred. Especially for the people who are familiar with the structure and hardware of the virtual machine.
- ***Infections***: A weak host system can easily be affected by infections. This usually happens when there are bugs in the operating system. If two or more virtual machines are connected with each other, the infections would spread to others as well.

Bibliography

- **StorySet for iCONS**
- **Wikipedia**
- **Microsoft**
- **IBM**

Acknowledgement

I have prepared this report under the supervision and guidance of ***Mr. Preetam Kumar Sur*** , Assistant Professor of Computer Science and Engineering department of Government College of Engineering & Textile Technology Serampore . I express my sincere gratitude to him for guiding me throughout this report making procedure and guiding me to understand the crucial concepts of ***Virtual Machine*** of ***Operating System***.



Signature of Student