Disadvantages of the Software-as-a-Service (SaaS):

- There may be legal reasons that preclude the use of off-premise or out-of-counry data storage.
- Security features of the SaaS Cloud Provider may not adequate for your needs.
- If you have a need for high-speed interaction between your internal software or software in another Cloud and the SaaS Cloud Provider, relying on an Internet connection may not provide the speed that you need.

3.4 PROS AND CONS OF CLOUD COMPUTING:

Part –A	1.List out some of the Advantages and Disadvantages in Cloud Computing

CLOUD COMPUTING: ADVANTAGES

1. Lower cost computers for users:
- No need for high powered computers
- The application runs not in the desktop.
- Client computers in cloud computing can be lower price , small hard disk, less memory.
- No software program has to be loaded.

2. Improved performance
- Lower IT infrastructure costs
- In a larger organization, the IT department could see lower cost after cloud computing.
- No need for large number of powerful servers.
- IT staff can use cloud to replace internal computing resources.

3.Fewer maintenance issues:
- Reduced maintenance cost.
- Less servers.
- Cloud applications.
- Less staff to maintain IT.

4. Lower software cost:
- No need to purchase separate software packages.
- Cost of installing and maintenance decrease.
- Rent the softwares.

5. Unlimited storage capacity:
- Virtually unlimited storage.
- Petabytes of data can be stored.

6. Increased computing power:
- The power of cloud is computing.
- Not limited to PC.
- Computing task and storage task can be done.

7. Instant software updates
- No high software update costs.

8. Increased data safety:
- No crash of hard disk.
- No data loss.
- Cloud is automatically duplicated.

9.Improved capability between OS:
- Work with any os. (windows or linux)

10. Improved document format capability:
- Document is be compatible with all formats.
- Eg: Word 2003, 2007

11.Easier group collaboration:
- Sharing documents.

DISADVANTAGES:

1. Can be slow:

- The Web based application can sometime slower than PC.
- The data is sent to the cloud makes slow.
- Cloud servers – slow.
- Internet – slow.

2. Features might be limited:

- Features of web based applications may be less.
- Eg: Google presentation

3. Stored data might not be secure

4. Platform Dependencies:

- Implicit dependency, also known as "vendor lock-in" is another of the disadvantages of cloud computing.

 Deep-rooted differences between vendor systems can sometimes make it impossible to migrate from one cloud platform to another.

 Not only can it be complex and expensive to reconfigure your applications to meet the requirements of a new host, but migration could also expose your data to additional security and privacy vulnerabilities.

5. Costs:

- Cloud computing – especially on a small scale and for short term projects – can be pricey. Though it can allow you to reduce staff and hardware costs, the overall price tag could end up higher than you expected.

6. Prone to Attack

 Storing information in the cloud could make your company vulnerable to external hack attacks and threats.

 As you are well aware, nothing on the Internet is completely secure and hence, there is always the lurking possibility of stealth of sensitive data.

3.5 IMPLEMENTATION LEVELS OF VIRTUALIZATION:

Part-A	**1.Give the role of a VM(AU/Nov/Dec2016)** **2.List the different types of Virtualization** **3.Define Virtualization** **4. What is the role of Hypervisor?** **5. .List the requirements of VMM(AU/Nov/Dec 2017)**
Part –B	**1.Discuss how virtualization is implemented in different layers(AU/April/May 2017)**

- Virtualization is software that separates physical infrastructures to create various dedicated resources. It is the fundamental technology that powers cloud computing
- "Virtualization software makes it possible to run multiple operating systems and multiple applications on the same server at the same time, It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware."
- The technology behind virtualization is known as a virtual machine monitor (VMM) or virtual manager, which separates compute environments from the actual physical infrastructure.
- Virtualization makes servers, workstations, storage and other systems independent of the physical hardware layer,

TYPES OF VIRTUALIZATION:

- Storage virtualization
- Server virtualization
- Operating system-level virtualization
- Network virtualization:
- Application virtualization

Storage virtualization - Storage virtualization is the amalgamation of multiple network storage devices into what appears to be a single storage unit. Storage virtualization is usually implemented via software applications and often used in SAN (storage area network), a high-speed sub network of shared storage devices, and makes tasks such as archiving, back-up, and recovery easier and faster.

Server virtualization - Server virtualization is the partitioning of a physical server into smaller virtual servers to help maximize your server resources. In server virtualization the resources of the server itself are hidden, or masked, from users, and software is used to divide the physical server into multiple virtual environments, called virtual or private servers.

Operating system-level virtualization - More commonly called OS-level virtualization. A type of server virtualization technology which works at the OS layer. The physical server and single instance of the operating system is virtualized into multiple isolated partitions, where each partition replicates a real server.

Network virtualization

- Network virtualization (NV) is using network resources through a logical segmentation of a single physical network.
- Network virtualization is achieved by installing software and services to manage the sharing of storage, computing cycles and applications.
- Network virtualization treats all servers and services in the network as a single pool of resources that can be accessed without regard for its physical components.
- The term network virtualization is often used to describe many things including network management, storage virtualization, and even grid computing.

Application virtualization

- Also called *application service virtualization*.
- **Application virtualization is layered on top of other virtualization technologies, such as storage virtualization or machine virtualization to allow computing resources to be** distributed dynamically in real time.
- In standard computing, applications install their settings onto the host operating system, hard-coding the entire system to fit that application's needs.
- With application virtualization, each application brings down its own set of configurations on-demand, and executes in a way so that it sees only its own settings.

LEVELS OF VIRTUALIZATION

- A traditional computer runs with a host operating system specially tailored for its hardware architecture, as shown in Figure 1(a).
- After virtualization, different user applications managed by their own operating systems (guest OS) can run on the same hardware, independent of the host OS. This is often done by adding additional software, called a *virtualization layer* as shown in Figure 1(b).

- This virtualization layer is known as *hypervisor* or *virtual machine monitor* (VMM). The VMs are shown in the upper boxes, where applications run with their own guest OS over the virtualized CPU, memory, and I/O resources.

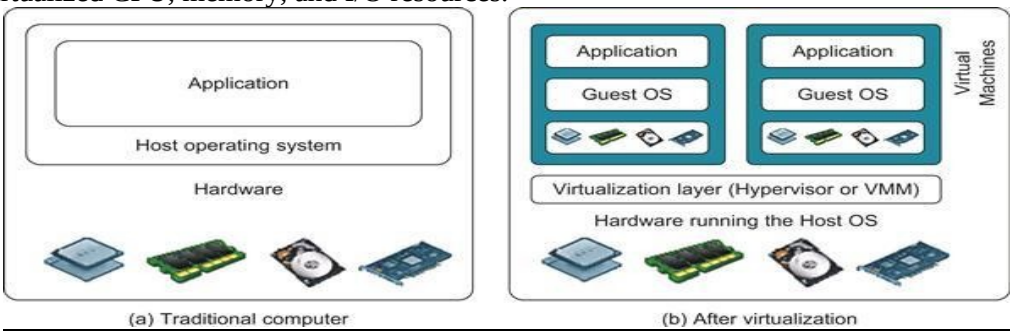


Figure 1: Architecture of a computer system before and after virtualization

- The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively.
- This can be implemented at various operational levels, as we will discuss shortly.
- The virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system. Common virtualization layers include the *instruction set architecture (ISA)* level, hardware level, operating system level, library support level, and application level.

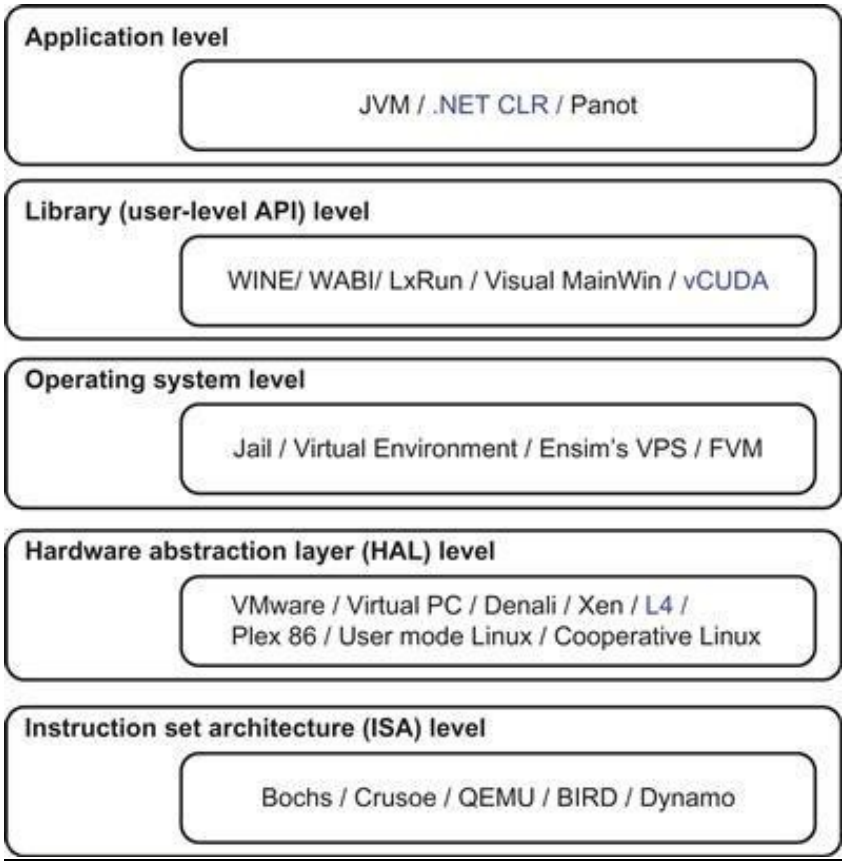


Figure 2: Virtualization ranging from hardware to applications in five abstraction levels.

Instruction Set Architecture Level

- For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.
- The basic emulation method is through code interpretation.
- An interpreter program interprets the source instructions to target instructions one by one.
- The performance, dynamic binary translation is desired.
- This approach translates basic blocks of dynamic source instructions to target instructions.
- The basic blocks can also be extended to program traces or super blocks to increase translation efficiency.
- Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

Hardware Abstraction Level

- Hardware-level virtualization is performed right on top of the bare hardware. On the one hand, this approach generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware through virtualization.
- The intention is to upgrade the hardware utilization rate by multiple users concurrently.
- The idea was implemented in the IBM VM/370 in the 1960s. More recently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS applications.

Operating System Level

- This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers.
- The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users. It is also used, to a lesser extent, in consolidating server hardware by moving services on separate hosts into containers or VMs on one server.

Library Support Level

- Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS.
- Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.
- The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.
- Another example is the vCUDA which allows applications executing within VMs to leverage GPU hardware acceleration.

User-Application Level

- Virtualization at the application level virtualizes an application as a VM. On a traditionalOS, an application often runs as a process. Therefore, application-level virtualization isalso known as process-level virtualization. The most popular approach is to deploy highlevel language (HLL) VMs.
- In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VMthat can run programs written and compiled to a particular abstract machine definition.

- Any program written in the HLL and compiled for this VM will be able to run on it.
- The result is an application that is much easier to distribute and remove from user workstations.
- An example is the LANDesk application virtualization platform which deploys software applications as self-contained, executable files in an isolated environment without requiring installation, system modifications, or elevated security privileges.

Relative Merits of Virtualization at Various Levels (More "X"'s Means Higher Merit, With a Maximum of 5 X's)

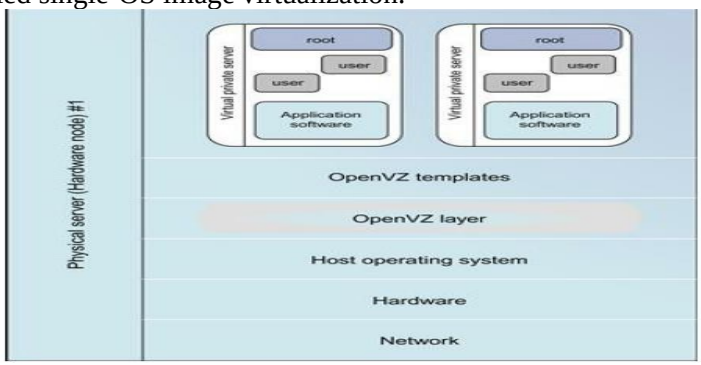
Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

Virtualization Support at the OS Level

- As cloud computing has at least two challenges.
- The first is the ability to use a variable number of physical machines and VM instances depending on the needs of a problem.
- For example, a task may need only a single CPU during some phases of execution but may need hundreds of CPUs at other times.
- The second challenge concerns the slow operation of instantiating new VMs.

Why OS-Level Virtualization?

- Besides slow operation, storing the VM images also becomes an issue.
- To reduce the performance overhead of hardware level virtualization, even hardware modification is needed.
- OS-level virtualization provides a feasible solution for these hardware-level virtualization issues.
- Operating system virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources.
- It enables multiple isolated VMs within a single operating system kernel. This kind of VM is called a virtual execution environment (VE), Virtual Private System (VPS), or simply container. From the user's point of view, VEs look like real servers.
- This means a VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings. Although VEs can be customized for different people, they share the same operating system kernel. Therefore, OS-level virtualization is also called single-OS image virtualization.



Advantages of OS Extensions

- Compared to hardware-level virtualization, the benefits of OS extensions are twofold:
- These benefits can be achieved via two mechanisms of OS-level virtualization:

- All OS-level VMs on the same physical machine share a single operating system kernel; and (2) the virtualization layer can be designed in a way that allows processes in VMs to access as many resources of the host machine as possible, but never to modify them.

Disadvantages of OS Extensions

- The main disadvantage of OS extensions is that all the VMs at operating system level on a single container must have the same kind of guest operating system.
- For example, a Windows distribution such as Windows XP cannot run on a Linux-based container.
- Some prefer Windows and others prefer Linux or other operating systems.

3.6 VIRTUALIZATION STRUCTURES/TOOLS AND MECHANISMS

Part –B	1.What is Virtualization? Describe para and Full virtualization architectures. Compare and Contrast them (AU/Nov/Dec 2017)

- In general, there are three typical classes of VM architecture.
- Before virtualization, the operating system manages the hardware.
- After virtualization, a virtualization layer is inserted between the hardware and the operating system.
- In such a case, the virtualization layer is responsible for converting portions of the real hardware into virtual hardware. Therefore, different operating systems such as Linux and Windows can run on the same physical machine, simultaneously.
- Depending on the position of the virtualization layer, there are several classes of VM architectures, namely the hypervisor architecture, para-virtualization, and host-based virtualization.
- The hypervisor is also known as the VMM (Virtual Machine Monitor). They both perform the same virtualization operations.

Hypervisor and Xen Architecture

- The hypervisor supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces.
- The hypervisor software its directly between the physical hardware and its OS. This virtualization layer is referred to as either the VMM or the hypervisor.
- The hypervisor provides hyper calls for the guest OS and applications. Depending on the functionality, a hypervisor can assume micro kernel architecture like the Microsoft Hyper-V Or it can assume a monolithic hypervisor architecture like the VMware ESX for server virtualization.
- A micro-kernel hypervisor includes only the basic and unchanging functions (such asphysical memory management and processor scheduling).
- The device drivers and other changeable components are outside the hypervisor.
- A monolithic hypervisor implements all the aforementioned functions, including those of the device drivers.
- Therefore, the size of the hypervisor code of a micro-kernel hypervisor is smaller than that of a monolithic hypervisor. Essentially, a hypervisor must be able to convert physical devices into virtual resources dedicated for the deployed VM to use.

The Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a micro-kernel hypervisor, which separates the policy from the mechanism.
- The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0, Xen does not include any device drivers natively.