Internal Assessment Test 1 – October 2024

Sub:	**Cloud Computing**	Sub Code:	**21CS72**	Branch :	**AInDS**

Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VII**	**OBE**

		Answer any FIVE Questions	**MARKS**	**CO**	**RBT**
Q 1	a	Define cloud computing. Draw a neat diagram, explain major development for cloud computing?	[2+3]	CO 1	L1
	b	Describe the characteristics of cloud computing.	[5]	CO 1	L1
Q 2		What advantages does cloud computing offer to enterprises? What are the major distributed computing technologies that led to cloud computing?	[6+4]	CO 1	L2
Q 3		With the neat diagram explain the cloud computing reference model. What are the two different concept that service oriented computing (SOC) introduced?	[6+4]	CO 1	L2
Q 4	a	Define virtualization. What are some of the key benefits of virtualization technologies in modern IT infrastructures?	[2+3]	CO 2	L1
	b	Highlight some of the major characteristics that virtualization technologies offer in today's IT infrastructures.	[5]	CO 2	L2
Q 5		Explain the different types or categories of virtualization at various levels. Create a clear and well-organized diagram to represent the taxonomy of virtualization and its applications.	[5+5]	CO 2	L2
Q 6		What advantages and disadvantages do programming language-level offer? How does application-level virtualization differ from hardware-level virtualization offer?	[5+5]	CO 2	L1

ANSWERS

1a. **Cloud Computing:** Cloud computing is a technology that delivers various computing services such as servers, storage, databases, networking, software, analytics, and intelligence over the internet, often referred to as "the cloud." This enables users and organizations to access and use resources on-demand without the need to own and manage physical infrastructure.

1b. The evolution of distributed systems into cloud computing was driven by advancements in virtualization, networking, and automation. Distributed systems introduced concepts like decentralization, resource sharing, scalability, and fault tolerance, but their complexity spurred the need for abstraction and automation.

Virtualization is a core technology for cloud computing because it allows multiple virtual machines (VMs) to run on a single physical server, efficiently utilizing hardware resources and enabling dynamic allocation based on demand. It abstracts the physical hardware, making it possible to isolate workloads, improve security, and maximize resource utilization, which reduces costs and enhances flexibility. The most common example of virtualization is hardware virtualization. Virtualization technologies are also used to replicate runtime environments for programs.

Web 2.0 contributed significantly to the growth of cloud computing by promoting interactive, user-driven platforms focused on collaboration, sharing, and dynamic content. Web 2.0 brings interactivity and flexibility into Web pages, providing enhanced user experience by gaining Web-based access to all the functions that are normally found in desktop applications.

2. (i) Large enterprises can offload some of their activities to cloud-based systems

Large organizations often have extensive IT workloads, including data storage, application hosting, and analytics. By moving these activities to cloud-based systems, they can:

- Reduce Costs: Eliminate the need to maintain expensive on-premises infrastructure.
- Enhance Efficiency: Focus their IT teams on strategic projects rather than routine maintenance.
- Improve Scalability: Handle fluctuating workloads dynamically without purchasing additional hardware.
  For example, they might store backup data on the cloud or run high-computational tasks like machine learning training in cloud environments.

(ii) Small enterprises and startups can afford to translate their ideas into business results more quickly without excessive upfront costs

Cloud computing provides startups and small businesses with:

- Cost Savings: No need to invest heavily in servers, storage, or data centers initially.
- Accessibility: Tools and platforms available on-demand, reducing setup time.
- Flexibility: Resources can be scaled as the business grows.
  For instance, a new app development startup can use cloud platforms for hosting and analytics without buying physical servers, accelerating their time-to-market.

(iii) System developers can concentrate on the business logic rather than dealing with the complexity of infrastructure management and scalability

With cloud services like Platform as a Service (PaaS), developers are provided with a ready-to-use environment, including pre-configured servers, databases, and frameworks. This allows them to:

- Focus on writing and optimizing code for their application instead of managing servers, networking, and storage.
- Automatically scale their applications based on demand without manual intervention.
  For example, developers building an e-commerce website can focus on features like payment integration rather than setting up and managing a database.

(iv) End users can have their documents accessible from everywhere and any device

Cloud storage services (e.g., Google Drive, Dropbox) enable users to store files in the cloud, providing:

- Convenience: Access files from laptops, smartphones, tablets, or any device with internet connectivity.
- Real-Time Collaboration: Multiple users can work on the same document simultaneously.

- Data Backup: Files are safer from device loss or damage since they are stored remotely.
  For example, a student can work on a project at home on their laptop and then continue seamlessly on their smartphone while commuting.

3. A fundamental characteristics of cloud computing is the capability to deliver, on demand, a variety of IT services that are quite diverse from each other. Cloud computing services offerings into three major categories: Infrastructure-as-a-service (IaaS), Platform-as-a-services (PaaS), and Software-as-service (SaaS)

Infrastructure-as-a-service solutions deliver infrastructure on demand in the form of virtual hardware, storage, and networking. Infrastructure as a Service (IaaS) is a cloud computing model that provides virtualized computing resources such as servers, storage, networks, and operating systems over the internet. It eliminates the need for organizations to own and manage physical infrastructure, offering on-demand scalability, flexibility, and a pay-as-you-go pricing model. Platform-as-a-service solutions are the next step in the stack. Platform as a Service (PaaS) is a cloud computing model that provides a ready-to-use environment for developers to build, test, deploy, and manage applications without worrying about the underlying infrastructure. At the top of the stack, Software-as-a-service is a cloud computing model that delivers software applications over the internet, allowing users to access them via a web browser without the need for installation, maintenance, or updates. SaaS providers manage the underlying infrastructure, including servers, storage, and security, while offering applications that are ready to use.

Service-Oriented Computing (SOC) introduces and diffuses two critical concepts that are also fundamental to cloud computing: Quality of Service (QoS) and Software-as-a-Service (SaaS).

Quality of Service (QoS) refers to the set of parameters and metrics used to define and manage the performance, reliability, and overall user experience of a service. In the context of cloud computing, QoS ensures that services meet specific performance standards, such as response times, availability, and throughput, which are essential for maintaining user satisfaction and ensuring that cloud resources are efficiently utilized.

Software-as-a-Service (SaaS), on the other hand, is a cloud computing model where software applications are provided over the internet as a service, rather than being hosted on a user's local infrastructure. In SOC, SaaS emphasizes the delivery of software that is centrally hosted, scalable, and accessible via web browsers, reducing the need for users to manage installations, updates, and infrastructure. This model makes it easier for businesses to access sophisticated software applications on-demand, with minimal upfront costs, and it aligns with the cloud's principles of flexibility, scalability, and cost-effectiveness.

4a. Virtualization: Virtualization is a technology that enables the creation of virtual versions of physical resources, such as servers, storage devices, networks, or operating systems. By using virtualization, a single physical machine can run multiple virtual machines (VMs), each with its own operating system and applications, effectively isolating them from one another.

- **Increased Performance and Computing Capacity**: Advances in hardware, such as faster processors, more memory, and higher storage capacities, have significantly improved computing performance. Virtualization leverages these improvements by allowing multiple virtual machines to run simultaneously on a single physical machine. This means that businesses can do more with less hardware, improving efficiency and reducing costs.

- **Underutilized Hardware and Software Resources**: Virtualization allows multiple virtual instances to run on a single physical machine, fully utilizing its computing power and improving the overall efficiency of the infrastructure. This helps reduce the waste of underused resources, leading to better cost management and resource allocation.

- **Lack of Space**: Virtualization reduces the need for numerous physical machines by consolidating workloads onto fewer, more powerful servers, thus minimizing the physical space required for IT infrastructure. This is particularly beneficial for businesses looking to scale without needing large physical spaces to accommodate the hardware.

- **Greening Initiatives**: Environmental sustainability has become a priority for many organizations, and virtualization plays a crucial role in "greening" IT infrastructures. By consolidating multiple workloads onto fewer physical servers, virtualization reduces the number of machines required, leading to lower energy consumption, less cooling needed, and a smaller carbon footprint. This contributes to more environmentally friendly operations, which is increasingly important for companies aiming to meet sustainability goals and reduce their environmental impact.

- **Rise of Administrative Costs**: As organizations grow, the complexity of managing and maintaining IT infrastructure increases, leading to higher administrative costs. Virtualization helps mitigate this by simplifying system management. With virtual machines, administrators can manage resources more easily, automate processes, and reduce the number of physical servers that need to be maintained.

4b. Characteristics of the virtualization

Increased Security: Virtualization enhances security by providing isolation between virtual machines (VMs), ensuring that if one VM is compromised, the attacker remains contained within that VM and cannot easily affect others or the host system. It enables segmentation of sensitive applications and data into separate virtual environments, reducing exposure to risks. Centralized management platforms allow administrators to monitor and secure all VMs from a single interface, ensuring timely patching and real-time threat detection. Virtualization also simplifies disaster recovery, enabling quick replication of VMs to secure environments, thus maintaining business continuity in the event of a security incident.

Managed Execution: Virtualization of the execution environment not only enhances security but also enables a broader range of features that improve system functionality and flexibility. Key features include:

Sharing: Virtualization allows multiple virtual machines to share the same physical hardware, leading to efficient resource utilization. This shared environment can be managed in a way that ensures each VM operates independently, with dedicated resources like CPU, memory, and storage, all while running on the same physical server.

Aggregation: Virtualization enables the aggregation of computing resources, allowing different systems or applications to pool their resources for greater performance and capacity.

Emulation: Virtual machines can emulate hardware, allowing software to run in a simulated environment, even if the underlying hardware differs. It allows software to be portable across different systems without requiring modifications.

Isolation: One of the key benefits of virtualization is the isolation of applications and environments. Each virtual machine runs independently, so issues or failures in one VM do not affect others.

Portability: Portability in virtualization refers to the seamless movement of virtual machines (VMs), applications, or workloads across different physical hosts, platforms, or environments without significant modifications. This capability ensures platform independence by decoupling software from hardware, allowing VMs to run on any compatible hypervisor or platform. It facilitates live migration for load balancing, maintenance, or disaster recovery, ensuring minimal downtime. Portability simplifies deployment by enabling preconfigured VMs to be replicated across development, testing, and production environments, ensuring consistency and efficiency.

5. Taxonomy of Virtualization techniques

Virtualization covers a wide range of emulation techniques that are applied to different areas of computing.

Execution virtualization: Execution virtualization includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer.

1. Machine reference model

Virtualizing an execution environment at different levels of the computing stack requires a reference model that defines clear interfaces between abstraction layers. Virtualization techniques replace one of these layers and intercept calls directed toward it, ensuring proper interaction with the underlying layer. Modern computing systems can be described using a layered reference model. At the base, the Instruction Set Architecture (ISA) serves as the interface between hardware and software, defining instructions, registers, and memory management for the processor. Above this, the Application Binary Interface (ABI) separates the operating system from applications and libraries, covering low-level details like data types, alignment, and system calls, enabling portability across operating systems with the same ABI. At the highest level, the Application Programming Interface (API) connects applications to the OS, translating high-level operations into machine-level instructions executed by the processor using resources like registers and memory.

2. Hardware level virtualization

Hardware virtualization is a technology that allows multiple operating systems or applications to run on a single physical machine by abstracting the underlying hardware into virtual machines (VMs). This is achieved using a software layer known as a hypervisor, which manages the allocation of physical resources, such as CPU, memory, and storage, to the VMs. The hypervisor creates and manages these virtual environments, enabling them to operate independently as if they were running on separate physical hardware.

3. Programing language level virtualization

Programming level virtualization abstracts the underlying hardware and operating system at the programming interface level. It allows developers to execute programs in a virtualized environment independent of the platform on which they were developed. This approach is commonly used in environments such as the Java Virtual Machine (JVM) or .NET Common Language Runtime (CLR), where applications are compiled to an intermediate bytecode and then executed on a virtual machine that interprets the bytecode. This ensures portability, security, and a consistent execution environment across various hardware and operating systems.

4. Application level virtualization

Application level virtualization encapsulates individual applications in a virtual container that includes all the dependencies and settings required for the application to run. This ensures that the application operates independently of the underlying operating system and other installed software. It simplifies deployment, reduces compatibility issues, and enables applications to be run on multiple devices without modification.

5. Network virtualization

Network virtualization abstracts the physical components of a network, such as switches, routers, and cables, to create virtual networks that operate independently of the underlying hardware. It allows multiple virtual networks to coexist on the same physical infrastructure, enabling greater flexibility, resource optimization, and enhanced security. Network virtualization supports features like virtual LANs (VLANs), software-defined networking (SDN), and virtual private networks (VPNs), which simplify network management, optimize traffic flow, and improve scalability.

6. Advantages of Programming Level Virtualization:

Portability:

Programs written for a virtualized environment can run on any underlying hardware or operating system that supports the virtualization layer, improving cross-platform compatibility.

Sandboxing:

It isolates applications from the underlying system, enhancing security by preventing untrusted code from affecting the host environment.

Ease of Development:

Provides developers with a consistent environment, reducing the complexity of managing different system dependencies and configurations.

Resource Optimization:

Enables efficient use of hardware resources by running multiple applications on a single system without interference.

Fault Tolerance:

Virtualization can create snapshots or backups of the virtual environment, making it easier to recover from failures.

Scalability:

Applications in a virtualized programming environment can easily scale by duplicating or migrating instances to meet demand.

Testing and Debugging:

Virtual environments can replicate different hardware and OS configurations, making it easier to test and debug applications in varied conditions.

Disadvantages of Programming Level Virtualization:

Performance Overhead:

The abstraction layer introduces additional processing overhead, which can reduce the performance of applications compared to running natively.

Complexity:

Managing and maintaining virtual environments can be complex, especially when integrating with existing systems or troubleshooting issues.

Limited Access to Hardware:

Applications in a virtualized environment may not have direct access to underlying hardware features, limiting their capabilities for certain use cases.

Dependency on the Virtualization Layer:

If the virtualization layer has bugs or vulnerabilities, it can affect all applications running within the environment.

Resource Contention:

Multiple virtualized environments sharing the same physical resources can lead to contention and degrade performance.

Compatibility Issues:

Some hardware or software features may not be supported by the virtualization layer, leading to limitations in application functionality.

Licensing and Costs:

Virtualization tools and software often come with licensing fees, and maintaining virtualized environments can increase operational costs.

Differences Between Hardware Level and Application Level Virtualization

Aspect	Hardware Level Virtualization	Application Level Virtualization
Definition	Virtualizes physical hardware to create multiple virtual machines (VMs) running their own OS instances.	Virtualizes the runtime environment to allow applications to run in isolated or emulated environments.
Granularity	Operates at the hardware layer, abstracting the physical machine.	Operates at the application layer, abstracting the application runtime.
Examples	VMware, VirtualBox, Hyper-V, KVM.	Docker, Kubernetes, JVM (Java Virtual Machine), .NET CLR.
Purpose	To enable multiple operating systems to run on a single physical machine.	To isolate and manage application environments or dependencies.
Performance Overhead	Higher, due to the need to emulate hardware and run separate OS instances.	Lower, as it focuses only on the application layer, not the entire OS.
Isolation	High isolation, as each VM runs its own OS and applications.	Moderate isolation; focuses on isolating applications but shares the host OS.
Resource Usage	Requires significant physical resources due to full OS virtualization.	More lightweight; uses fewer resources since it virtualizes only applications.
Setup Complexity	More complex to set up and maintain because it involves virtualizing entire systems.	Easier to set up and manage; focuses on specific application environments.
Use Cases	- Server consolidation- Development and testing of OS-dependent software- Running legacy systems	- Containerized applications- Microservices- Cross-platform application portability
Scalability	Less scalable due to resource-intensive operations and VM overhead.	Highly scalable, especially for containerized solutions like Docker.
Dependency on Host OS	Independent of the host OS, as each VM can have its own OS.	Dependent on the host OS, as containers or runtimes share the host's kernel.
Startup Time	Slower, as it involves booting an entire OS.	Faster, as only the application or runtime environment is initialized.