USN | | | | | | | | | | |

Internal Assessment Test 1 – March 2025

Sub:	**Cloud Computing & Security**	Sub Code:	**BIS613D**	Branch:	**AIML & CSE-AIML**

Date:	**26/03/25**	Duration:	**90 minutes**	Max Marks:	**50**	Sem/Sec:	**VI**	**OBE**

		Answer any FIVE FULL Questions	**MARKS**	**CO**	**RBT**
1	a	Distinguish between full virtualization and para virtualization.	[10]	2	L2
2	a	How would you apply the concept of virtualization to meet the requirements at the implementation level? Explain the steps and considerations involved in this process.	[10]	2	L3
3	a	How would you implement CPU virtualization and memory virtualization in a virtualized environment?	[10]	2	L3
4	a	Compare and explain the different types of cloud computing models, highlighting their advantages and limitations.	[10]	2	L2
5	a	List the applications of HPC & HTC systems.	[05]	1	L1
	b	Explain Multicore CPU with a neat diagram	[05]	1	L2
6	a	Explain the structure of layered cloud architecture and how each layer interacts with the others to deliver cloud services.	[10]	3	L2

Q1(a) Distinguish between Full Virtualization and Para Virtualization

Virtualization enables multiple operating systems (OS) to run on a single physical machine by abstracting hardware resources. **Full virtualization** and **para-virtualization** are two types of virtualization that differ in how they interact with the hypervisor and the underlying hardware.

Full Virtualization

Full virtualization allows an unmodified guest OS to run on a virtualized hardware environment. The hypervisor (Virtual Machine Monitor - VMM) **completely emulates the hardware**, making the guest OS unaware that it is running in a virtualized environment.

Key Features of Full Virtualization:

1. **Binary Translation** - The hypervisor intercepts and translates privileged instructions before execution.
2. **Unmodified Guest OS** - The OS does not need modifications because the hypervisor emulates real hardware.
3. **Requires Hardware Support** - Hardware-assisted virtualization (Intel VT-x, AMD-V) improves performance by reducing emulation overhead.
4. **Performance Overhead** - Due to binary translation and hardware emulation, it is slightly slower than para-virtualization.

Advantages of Full Virtualization:

- Allows running **any OS** without modification.
- Strong **isolation** between VMs.
- Supports **legacy OS** that cannot be modified.

Disadvantages of Full Virtualization:

- **Higher overhead** due to binary translation.
- **Requires hardware acceleration** for better performance.
- Can be **resource-intensive**.

Para Virtualization

In para-virtualization, the guest OS is **aware that it is running in a virtualized environment** and is **modified to communicate directly** with the hypervisor. This reduces the overhead caused by binary translation.

Key Features of Para Virtualization:

1. **No Full Hardware Emulation** - The OS interacts with the hypervisor via hypercalls.
2. **Modified Guest OS** - The OS is optimized for the hypervisor.
3. **Better Performance** - Since the guest OS cooperates with the hypervisor, there is **less overhead** than in full virtualization.
4. **No Need for Hardware Acceleration** - It can run on older hardware without VT-x or AMD-V.

Advantages of Para Virtualization:

- **Faster execution** due to reduced emulation.
- **Lower resource consumption** compared to full virtualization.
- **Better performance** for applications with high system calls.

Disadvantages of Para Virtualization:

- **Requires OS modification** (Not all OS support para-virtualization).
- Less flexibility compared to full virtualization.
- **Not suitable for closed-source OS** like Windows (as it cannot be modified).

Comparison Table

Feature	Full Virtualization	Para Virtualization
Guest OS Modification	Not required	Required
Performance	Slower (higher overhead)	Faster (lower overhead)

Feature	Full Virtualization	Para Virtualization
Hardware Requirement	Requires VT-x / AMD-V for better performance	Can work without hardware acceleration
Use Case	Running any OS without changes	Running open-source OS with modifications
Examples	VMware Workstation, VirtualBox, Microsoft Hyper-V	Xen, VMware ESXi (with para-virtualized drivers)

Q2(a) How would you apply the concept of virtualization to meet the requirements at the implementation level? Explain the steps and considerations involved in this process.

Virtualization is used to **increase efficiency, resource utilization, and scalability** in IT environments. Implementing virtualization requires a structured approach.

1. Identifying Requirements

- Determine the need for virtualization (server consolidation, testing, cloud computing).
- Assess **hardware compatibility** (CPU, RAM, disk space).
- Choose between **full, para, or OS-level virtualization**.

2. Selecting the Virtualization Type

Virtualization Type	Description	Use Case
Full Virtualization	Emulates hardware completely	Running unmodified OS
Para Virtualization	Guest OS modified to interact with the hypervisor	Performance-critical applications
OS-Level Virtualization	Containers share the same OS kernel	Lightweight application deployment (e.g., Docker)

3. Choosing a Hypervisor

Hypervisor Type	Description	Examples
Type 1 (Bare Metal)	Installed directly on hardware	VMware ESXi, Microsoft Hyper-V, Xen
Type 2 (Hosted)	Runs on an existing OS	VirtualBox, VMware Workstation

4. Hardware Configuration

- Enable **VT-x / AMD-V** in BIOS for CPU virtualization.
- Allocate **sufficient RAM** to avoid swapping.
- Configure **network settings** (bridged, NAT, host-only).

5. Installing the Hypervisor

- Install **VMware ESXi, Xen, or KVM** on bare metal for enterprise use.
- Install **VirtualBox or VMware Workstation** for personal use.

6. Creating Virtual Machines

- Allocate CPU, memory, storage for each VM.
- Install guest OS.
- Configure **snapshots and backups**.

7. Performance Optimization

- Enable **memory ballooning** to dynamically allocate memory.
- Use **CPU pinning** to assign cores.
- Enable **storage deduplication** to reduce disk usage.

8. Security Considerations

- Isolate VMs using **firewalls**.
- Implement **access controls** (role-based access).
- Regularly update **hypervisor and guest OS**.

9. Monitoring and Management

- Use **VMware vSphere, Microsoft SCVMM** for monitoring.
- Automate **backup and recovery**.

Q3(a) How would you implement CPU virtualization and memory virtualization in a virtualized environment?

CPU Virtualization

CPU virtualization allows multiple **virtual CPUs (vCPUs)** to share a physical CPU.

Steps to Implement CPU Virtualization

1. **Hypervisor Installation** – A hypervisor (VMware, Xen, KVM) manages vCPUs.
2. **CPU Scheduling** – The hypervisor assigns CPU cycles to VMs.
3. **Hardware Support (VT-x, AMD-V)** – Reduces instruction emulation overhead.
4. **Time-Slicing Mechanism** – Allows fair CPU time allocation.
5. **Performance Optimization** – Uses CPU pinning to assign specific cores to VMs.

Memory Virtualization

Memory virtualization allows multiple VMs to share **physical memory (RAM)** efficiently.

Techniques for Memory Virtualization

1. **Memory Overcommitment** – Allocates more virtual memory than available.
2. **Ballooning** – Reallocates memory dynamically between VMs.
3. **Transparent Page Sharing (TPS)** – Identifies duplicate memory pages and merges them.
4. **Swapping** – Moves inactive pages to disk storage.
5. **NUMA Awareness** – Optimizes memory access latency.

Q4(a) Compare and explain the different types of cloud computing models, highlighting their advantages and limitations.

Cloud computing models define how services are delivered over the internet. The three main cloud models are **IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service).**

1. Infrastructure as a Service (IaaS)

IaaS provides **virtualized computing resources** like servers, storage, and networking. Users have full control over their infrastructure without maintaining physical hardware.

Key Features:

- Provides **virtual machines, storage, networking**.
- Users manage **OS, applications, and configurations**.
- Uses a **pay-as-you-go** pricing model.

Examples:

- **Amazon Web Services (AWS) EC2**
- **Google Compute Engine (GCE)**
- **Microsoft Azure Virtual Machines**

Advantages of IaaS:

✓ **Scalability:** Resources can be increased/decreased as needed.
✓ **Cost-Effective:** No upfront hardware investment.
✓ **Flexibility:** Users control software and configurations.

Limitations of IaaS:

✗ **Requires IT expertise** for setup and management.
✗ **Security risks** as data is stored in a shared environment.
✗ **Performance dependency** on network latency.

---

2. Platform as a Service (PaaS)

PaaS provides a **pre-configured development environment** for developers to build, test, and deploy applications without managing infrastructure.

Key Features:

- Provides **runtime environment, databases, and middleware**.
- Developers focus on coding, not infrastructure management.
- Automates scaling and load balancing.

Examples:

- **Google App Engine**
- **Microsoft Azure App Services**
- **AWS Elastic Beanstalk**

Advantages of PaaS:

✓ **Faster development** due to pre-configured environments.
✓ **Automatic scaling** to handle traffic spikes.
✓ **Less maintenance** as infrastructure is managed by the provider.

Limitations of PaaS:

✗ **Limited customization** compared to IaaS.
✗ **Vendor lock-in**, making it hard to migrate to another provider.
✗ **Security risks**, as data is controlled by a third party.

---

3. Software as a Service (SaaS)

SaaS provides **ready-to-use software applications** over the internet. Users access applications through a browser without installation.

Key Features:

- No need for **installation or updates**.
- Users access software via **subscriptions**.
- Multi-tenant architecture (one instance serves multiple users).

Examples:

- **Google Workspace (Docs, Sheets, Gmail)**
- **Microsoft 365 (Word, Excel, Teams)**
- **Salesforce CRM**

Advantages of SaaS:

✓ **Easy access** from any device with an internet connection.
✓ **No maintenance** required by users.
✓ **Scalable pricing models** based on usage.

Limitations of SaaS:

✗ **Limited control** over configurations.
✗ **Data privacy concerns**, as data is stored in the cloud.
✗ **Internet dependency**, meaning no access if the network is down.

Comparison Table:

Feature	**IaaS**	**PaaS**	**SaaS**
Control	Full control over OS & apps	Control over code & apps	No control, only user access
Use Case	Hosting virtual machines	Developing & deploying apps	Using cloud-based apps
Technical Expertise	High	Medium	Low
Scalability	High	High	Limited
Example	AWS EC2	Google App Engine	Google Docs

Q5(a) List the applications of HPC & HTC systems.

High-Performance Computing (HPC)

HPC systems process large amounts of data **in parallel** using powerful clusters.

Applications of HPC:

1. **Weather Forecasting** – Simulating climate patterns.
2. **Molecular Modeling** – Used in drug discovery and genetics.
3. **Financial Modeling** – Risk analysis and stock market predictions.
4. **Artificial Intelligence & Machine Learning** – Training large AI models.
5. **Engineering Simulations** – Used in aerospace, automotive, and nuclear physics.

High-Throughput Computing (HTC)

HTC focuses on running many **small, independent tasks** over time.

Applications of HTC:

1. **Big Data Analytics** – Processing large datasets.
2. **Cloud Computing** – Running distributed workloads.
3. **Cryptocurrency Mining** – Running blockchain calculations.

4. **Video Rendering** – Used in CGI animation and movie production.
5. **Genomic Sequencing** – Used in bioinformatics for DNA analysis.

Q5(b) Explain Multicore CPU with a neat diagram

A **multicore CPU** consists of multiple processing units (**cores**) on a single chip, allowing **parallel execution of tasks**.

Key Features of a Multicore CPU:

- **Multiple cores** allow simultaneous execution of tasks.
- **Shared cache** improves memory access efficiency.
- **Lower power consumption** compared to multiple single-core CPUs.

Diagram of a Multicore CPU:

```
---------------------------------------
|     Multicore Processor (CPU)      |
| ----------------------------------- |
| Core 1 | Core 2 | Core 3 | Core 4 |
| Cache  | Cache  | Cache  | Cache  |
| -------- | -------- | -------- | -------- |
|          Shared Memory           |
---------------------------------------
```

Advantages of Multicore CPUs:

✓ **Increased Performance** – Can execute multiple threads in parallel.
✓ **Energy Efficient** – Reduces power consumption per operation.
✓ **Better Multitasking** – Runs multiple applications smoothly.

Limitations of Multicore CPUs:

✗ **Software Optimization Required** – Not all programs are optimized for multiple cores.
✗ **Heat Generation** – More cores lead to higher temperatures.
✗ **Diminishing Returns** – Adding more cores does not always improve performance.

Q6(a) Explain the structure of layered cloud architecture and how each layer interacts with the others to deliver cloud services.

The **layered cloud architecture** organizes cloud computing services into different levels to ensure seamless delivery.

Cloud Architecture Layers:

1. **Physical Layer (Infrastructure Layer)**
   - **Includes:** Servers, storage, networking devices.
   - **Function:** Provides **computing power** for cloud services.
2. **Virtualization Layer**
   - **Includes:** Hypervisors (VMware, KVM, Xen).
   - **Function:** Abstracts physical hardware to create virtual machines.
3. **Infrastructure Layer (IaaS)**
   - **Includes:** Compute, storage, networking as a service.
   - **Function:** Provides **on-demand virtual resources**.
4. **Platform Layer (PaaS)**
   - **Includes:** Databases, development frameworks (Node.js, Java).
   - **Function:** Supports application development and deployment.
5. **Application Layer (SaaS)**
   - **Includes:** Google Workspace, Microsoft 365.
   - **Function:** Delivers **ready-to-use applications** to end users.

Interaction Between Layers

- **Physical → Virtualization:** Hardware is abstracted by hypervisors.
- **Virtualization → IaaS:** Virtual machines are created and allocated to users.
- **IaaS → PaaS:** Developers build applications on the cloud infrastructure.
- **PaaS → SaaS:** End users access applications through a browser.