CBCS SCHEME

USN | | | | | | | | | |

BIS613D

Sixth Semester B.E./B.Tech. Degree Examination, June / July 2026
Cloud Computing and Security

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Explain the Platform Evolution of different computer technologies with a neat diagram.	10	L2	CO1
	b.	Outline eight reasons to adopt the cloud for upgraded internet applications and web services.	08	L2	CO1
	c.	Briefly explain Message Passing Interface (MPI).	02	L2	CO1
		OR			
Q.2	a.	Summarize all VM primitive operations with relevant diagram.	10	L2	CO1
	b.	Illustrate various system attacks and network threats to the cyber space, resulting in 4 types of losses with neat diagram.	10	L2	CO1
		Module – 2			
Q.3	a.	Demonstrate the architecture of a computer system before and after virtualization.	10	L2	CO2
	b.	Compare physical versus virtual cluster.	10	L2	CO2
		OR			
Q.4	a.	Construct the Live migration process of a VM from one host to another.	10	L3	CO2
	b.	Develop a architecture of live wire for intrusion detection using a dedication VM.	10	L3	CO2
		Module – 3			
Q.5	a.	Outline six design objectives for cloud computing.	06	L2	CO3
	b.	With neat diagram, build a cloud ecosystem with private cloud.	08	L3	CO3
	c.	Organize functional modules of GAE.	06	L2	CO3
		OR			
Q.6	a.	Identify basic requirements for managing the resources of a data center.	08	L3	CO3
	b.	Summarize six open challenges in cloud architecture development.	08	L2	CO3
	c.	Write a note on AWS.	04	L2	CO3

Module – 4

Q.7	a.	Demonstrate surfaces of attacks in a cloud computing environment with neat diagram.	08	L2	CO4
	b.	List out the top cloud security threats of CSA2016.	06	L1	CO4
	c.	Select four widely accepted fair information that " Consumer oriented commercial websites that collect personal identifying information from or about consumers online would be required to comply with.	06	L3	CO4
		OR			
Q.8	a.	Summarize the design goals of X_{gan}.	06	L2	CO4
	b.	Explain mobile devices and cloud security.	06	L2	CO4
	c.	Model an overview of reputation system design options.	08	L3	CO4
		Module – 5			
Q.9	a.	Outline important cloud platform capabilities.	10	L2	CO5
	b.	Organize the steps involved in Map Reduce.	10	L3	CO5
		OR			
Q.10	a.	Explain with a neat diagram how data flows if running a Map Reduce job at various task trackers using the hadoop library	10	L2	CO5
	b.	Explain data mutation sequence in GFS with diagram.	10	L2	CO5

* * * * *

Cloud Computing (Jun-July 2026)

1a.Explain the Platform Evolution of Different Computing Technologies with a Neat Diagram

Introduction

Computing technologies have evolved from centralized systems to cloud and ubiquitous computing. This evolution has improved **performance, scalability, resource sharing, accessibility, and efficiency**.

1. Early Computing Systems (1960–1980)

- Computing resources were mainly **centralized**.
- Lower-cost **minicomputers** such as:
  - DEC PDP-11
  - VAX Series
- Users accessed computing power through terminals connected to a central machine.

Features

- Centralized resource management
- Limited user access
- Expensive hardware

2. Personal Computer Era (1970–1990)

- Development of **VLSI microprocessors** led to personal computers.
- Computing became affordable and accessible to individuals.

Features

- Desktop computing
- Improved user productivity
- Independent computing systems

3. Portable and Network Computing (1980–2000)

- Rapid growth of portable computers and pervasive devices.
- Wired and wireless communication technologies became common.

Features

- Laptop and mobile computing
- Internet connectivity
- Network-based applications

4. Modern Computing Era (Since 1990)

- Emergence of:

    - High-Performance Computing (HPC)
    - High-Throughput Computing (HTC)
    - Clusters
    - Grids
    - Internet Clouds
- Cloud computing enabled **on-demand services over the Internet**.

Features

- Massive scalability
- Resource sharing
- Utility computing model

HPC and HTC Evolution

High-Performance Computing (HPC)

- Focuses on achieving maximum computational speed.
- Traditional supercomputers replaced by clusters of cooperative computers.
- Performance evolved from:
    - Gflops → Tflops → Pflops

Applications

- Scientific simulations
- Weather forecasting
- Engineering design

High-Throughput Computing (HTC)

- Focuses on completing a large number of tasks over time.
- Supports millions of users simultaneously.

Applications

- Search engines
- Web services
- Data analytics

Emerging Technologies

1. Peer-to-Peer (P2P) Networks

- Distributed file sharing and content delivery.
- Uses globally distributed client machines.

Examples

- BitTorrent
- Skype (early versions)

2. Virtualization

- Multiple Virtual Machines (VMs) run on the same hardware.
- Foundation of cloud computing and service-oriented computing.

Benefits

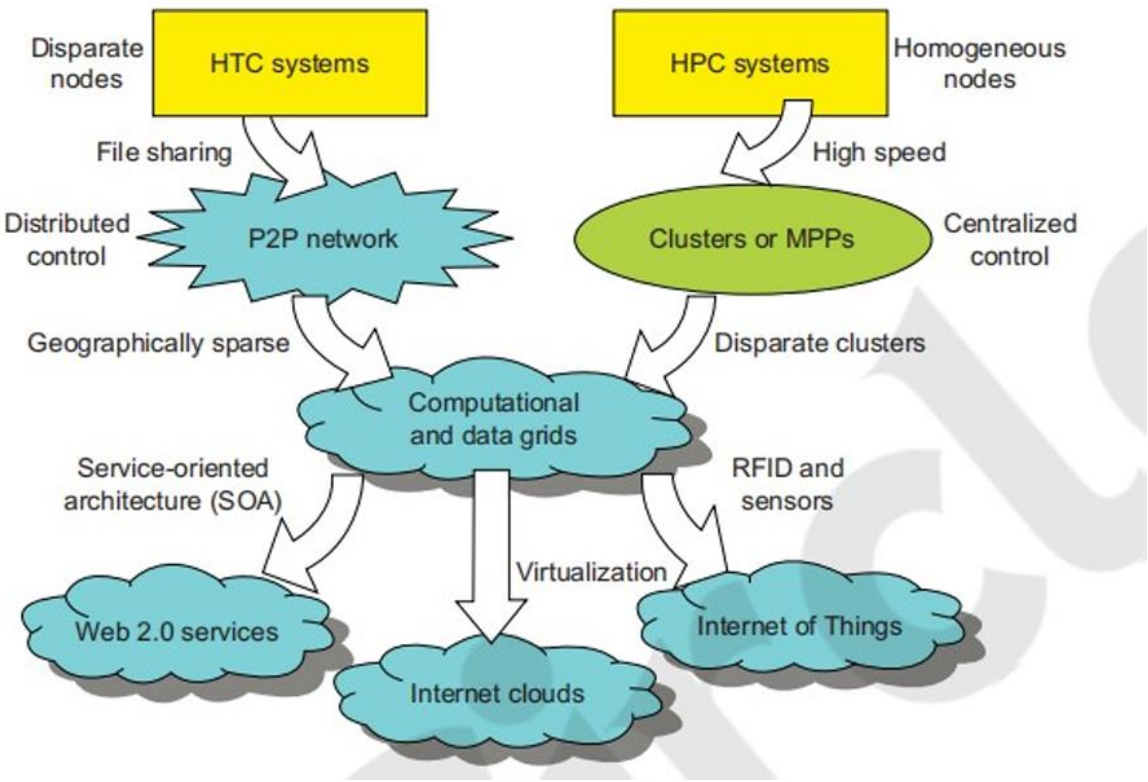
- Better resource utilization
- Isolation and flexibility

3. Internet of Things (IoT)

- Uses:
  - RFID
  - GPS
  - Sensors
- Connects physical objects through networks.

Applications

- Smart homes
- Smart cities
- Healthcare monitoring



1b. Outline 8 reasons to adapt the cloud for upgraded internet applications and web services.

Cloud computing is widely adopted for Internet applications and web services because it provides cost-effective, scalable, reliable, and on-demand computing resources.

1. Cost Effectiveness

- Reduces the need to purchase expensive hardware and software.
- Users pay only for the resources they use (pay-as-you-go model).
- Maintenance costs are lower than traditional systems.

2. Scalability

- Storage and computing power can be increased or decreased easily.
- Supports growing workloads without replacing infrastructure.

3. Resource Sharing

- Multiple users can share computing resources efficiently.
- Improves utilization of servers, storage, and networks.

4. Flexibility and On-Demand Access

- Services can be accessed anytime and from anywhere through the Internet.
- Resources can be provisioned whenever required.

5. Reduced Infrastructure Maintenance

- Cloud providers handle hardware management, updates, and maintenance.
- Developers can focus on application development rather than infrastructure management.

6. Support for Multi-user Applications

- Many users can access the same application simultaneously through web browsers.
- Enables online collaboration and web-based services.

7. Improved Security and Reliability

- Cloud data centers provide secure environments.
- Backup and disaster recovery mechanisms improve service availability.

8. Elastic and Self-Recovering Computing Power

- Resources are dynamically allocated during peak workloads.
- Distributed servers and virtualization enable fast failure recovery.

1c. **Briefly explain message passing interface(MPI)**

Message Passing Interface (MPI) is a standard programming model used for **parallel computing in distributed systems**. It enables multiple processes running on different machines or processors to communicate by sending and receiving messages.

Explanation

1. Standard for Parallel Programming

- MPI is widely used for developing parallel applications in distributed computing environments.
- It allows multiple processors to work together to solve problems faster.

2. Message-Based Communication

- Processes exchange data through messages.
- Communication is performed using explicit **send** and **receive** operations.

3. Process Coordination

- MPI manages communication among multiple processes.
- Each process performs a portion of the task and shares results with others.

4. High-Performance Computing (HPC)

- MPI is extensively used in HPC applications.
- Suitable for:
    - Scientific simulations
    - Engineering computations
    - Large-scale data processing

5. Distributed Memory Model

- Each processor has its own local memory.
- Data is exchanged through messages rather than shared memory.

6. Scalability

- MPI programs can run on a small number of processors or thousands of processors.
- Performance can improve as more nodes are added.

7. Explicit Communication

- The programmer controls when data is sent and received.
- Provides greater control over execution and performance.

8. Efficient Parallel Execution

- Large tasks are divided into smaller subtasks.
- Subtasks execute simultaneously, reducing overall execution time.

Key Features of MPI

- Supports **point-to-point communication**
- Supports **process synchronization**

- Supports **data exchange between distributed processes**
- **Portable and scalable** across clusters and distributed systems

2a. Summarize VM Primitive operations with relevant diagram.

Virtual Machine (VM) Primitive Operations are the basic operations used to create, manage, move, and control virtual machines in a virtualized environment. These operations improve the flexibility and efficiency of cloud computing.

1. Multiplexing

- Multiple virtual machines run on a single physical machine.
- Each VM has its own:
  - Application (App)
  - Operating System (OS)
- All VMs share the same hardware through a **Virtual Machine Monitor (VMM) / Hypervisor**.
- Improves hardware utilization and reduces operational cost.

2. Suspension (Storage)

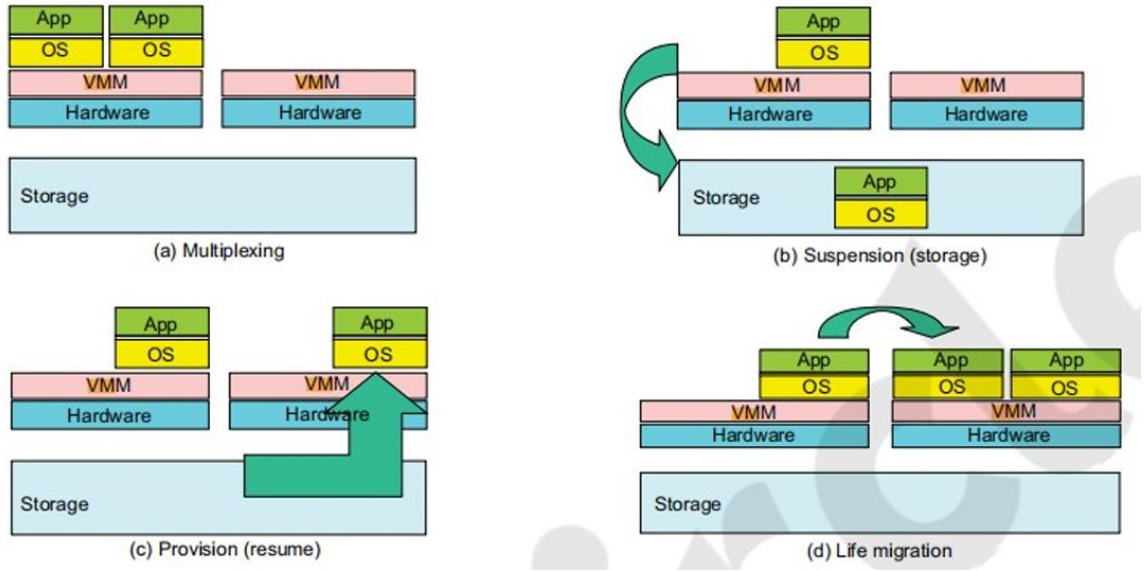
- Suspension means temporarily stopping a VM.
- The current state of the VM is saved to storage.
- CPU execution stops.
- Memory contents and execution state are preserved.
- The VM can later continue from the same point.

3. Provision / Resume

- Resume means restarting a suspended VM.
- The saved VM image is loaded from storage into hardware.
- The VM continues execution from where it was paused.
- Useful when additional resources are needed.

4. Live Migration

- Live migration transfers a running VM from one physical server to another **without shutting it down**.
- Applications continue running during migration.
- Used for:
  - Load balancing
  - Fault tolerance
  - Server maintenance



2b. Illustrate various system attacks and network threats to the cyberspace,resulting in 4 types of loses with neat diagram.

Security threats can affect the three major security goals: **Confidentiality, Integrity, and Availability (CIA)**, as well as authentication mechanisms.

1. Loss of Confidentiality

Confidentiality means protecting information from unauthorized access or disclosure.

Threats Causing Loss of Confidentiality

a) Eavesdropping

- Attackers secretly listen to network communications.
- Sensitive information may be intercepted.

b) Traffic Analysis

- Attackers analyze communication patterns and network traffic.
- Useful information can be gathered even without reading the actual data.

c) Media Scavenging

- Attackers recover deleted or sensitive data from storage devices.
- Confidential information may be exposed.

Effects

- Data leakage
- Privacy loss
- Unauthorized disclosure of sensitive information

2. Loss of Integrity

Integrity ensures that data remains accurate, correct, and unmodified.

Threats Causing Loss of Integrity

a) Penetration Attacks

- Attackers gain unauthorized access and alter data.

b) Trojan Horses

- Malicious programs disguised as legitimate software.
- Can modify, delete, or damage files.

c) Unauthorized Access

- Unauthorized users may change or corrupt data.

Effects

- Incorrect records and databases
- Loss of trust in data
- Improper system operation

3. Loss of Availability

Availability ensures that systems, services, and data are accessible whenever needed.

Threats Causing Loss of Availability

a) Denial of Service (DoS) Attacks

- Attackers flood a server or network with excessive requests.
- Legitimate users cannot access services.

b) Resource Exhaustion

- Excessive use of CPU, memory, bandwidth, or storage.
- Causes system slowdown or shutdown.

Effects

- Service interruption
- Website or application downtime
- Reduced productivity

4. Improper Authentication

Authentication verifies the identity of users before access is granted.

Problems Caused by Improper Authentication

a) Theft of Resources

- Attackers misuse computing resources and services.

b) Replay Attacks

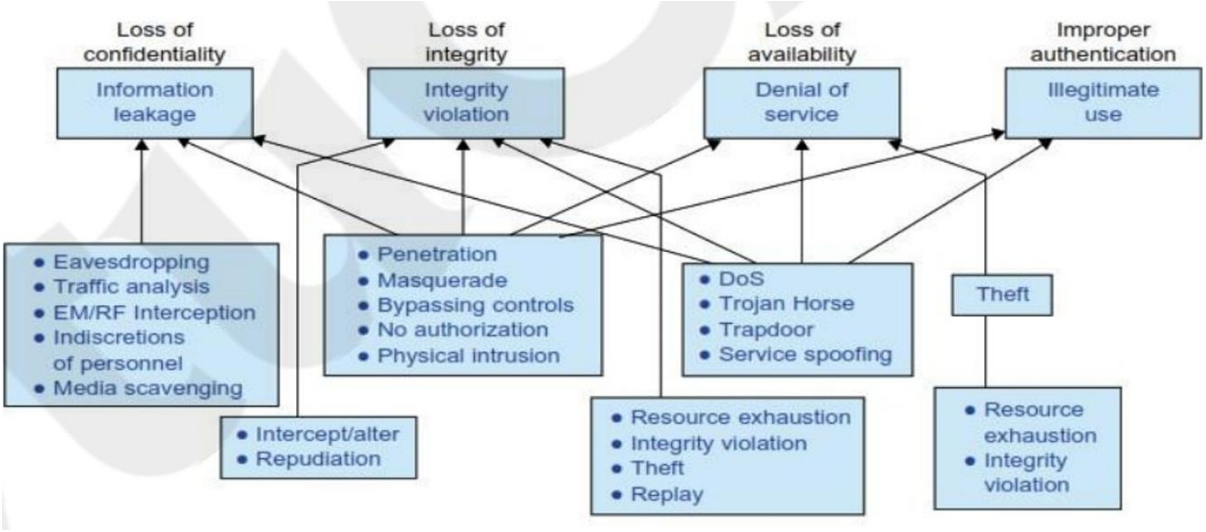
- Attackers capture login/session data and reuse it later.

c) Unauthorized Data Modification

- Attackers alter files or system settings without permission.

Effects

- Unauthorized access
- Resource misuse
- Security breaches



3a. Demonstrate the architecture of a computer system before and after virtualisation

1. Architecture Before Virtualization (Traditional Computer)

Explanation

1. A traditional computer consists of **Hardware, Host Operating System, and Applications**.
2. The operating system runs directly on the physical hardware.
3. Applications execute on top of the host operating system.
4. Only one operating system can run on a physical machine.
5. Hardware resources are dedicated to a single OS and its applications.
6. Resource utilization is often low because many resources remain underutilized.
7. Adding new services usually requires additional physical machines.

2. Architecture After Virtualization

Explanation

1. A **Virtualization Layer (Hypervisor/VMM)** is inserted between hardware and virtual machines.
2. The virtualization layer abstracts physical hardware resources.
3. Multiple virtual machines can run simultaneously on the same physical machine.
4. Each VM contains its own **Guest Operating System** and applications.
5. Different operating systems can run independently on the same hardware.
6. CPU, memory, storage, and network resources are shared among VMs.
7. Each VM is isolated from other VMs, improving reliability and security.
8. Resources can be allocated dynamically according to workload requirements.
9. Virtual machines can be easily created, cloned, migrated, and managed.
10. Hardware utilization is significantly improved.

Role of the Virtualization Layer (Hypervisor/VMM)

The **Virtualization Layer** acts as an interface between physical hardware and virtual machines.

Functions

1. Abstraction

- Hides hardware complexity from virtual machines.

2. Isolation

- Prevents one VM from affecting another VM.

3. Resource Allocation

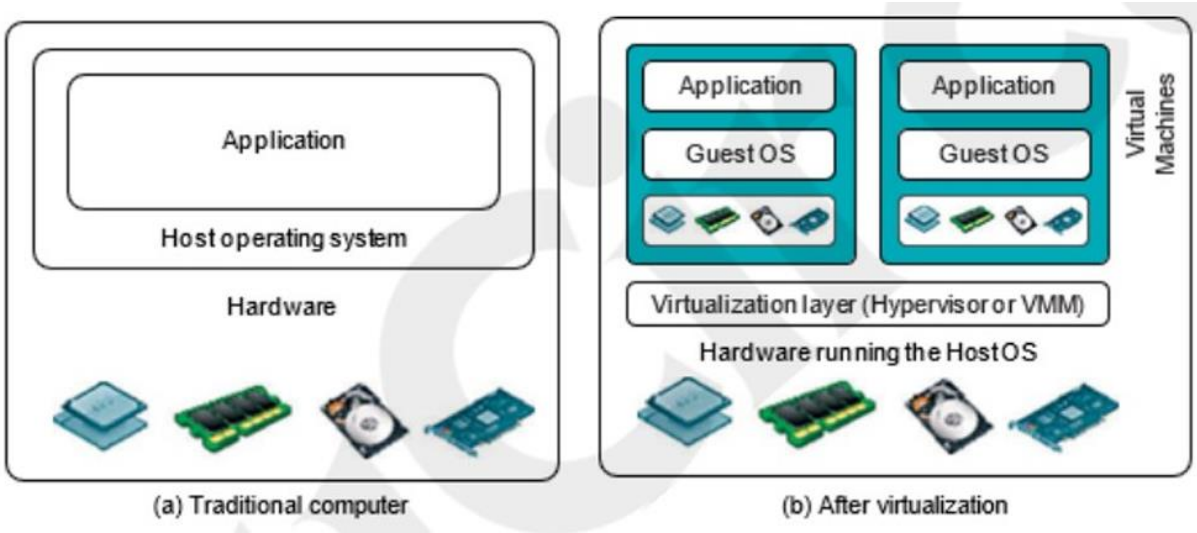
- Distributes CPU, memory, storage, and I/O resources among VMs.

4. Compatibility

- Supports multiple operating systems on the same hardware.

5. Portability

- Allows virtual machines to be moved between physical hosts.



3b. Compare physical and virtual clusters.

Physical Clusters	Virtual Clusters
1. Consist of interconnected **physical servers**.	1. Consist of interconnected **virtual machines (VMs)**.
2. Each node is a dedicated physical machine.	2. Each node is a virtual machine running on a physical host.
3. Resources are fixed and tied to hardware.	3. Resources can be dynamically allocated.
4. Hardware utilization is generally low.	4. Hardware utilization is high due to resource sharing.
5. Scalability requires adding new physical machines.	5. Scalability is achieved by creating new VMs.
6. Higher hardware, power, and maintenance costs.	6. Lower infrastructure and operational costs.
7. Difficult to migrate workloads between servers.	7. Supports VM migration and load balancing.
8. Recovery from failures is slower.	8. Faster recovery using VM snapshots and replication.
9. Limited flexibility and configurability.	9. Highly flexible and configurable.
10. Management is hardware-oriented.	10. Management is VM-oriented and automated.

4a. Construct the live migration process of a VM from one host to another.

Live Migration is the process of moving a running Virtual Machine (VM) from one physical host to another with **minimal downtime and service interruption**. It is commonly used for **load balancing, fault tolerance, server maintenance, and resource management** in cloud environments.

Steps in Live VM Migration

1. Pre-Migration

- Source host selects the VM for migration.
- Destination host is chosen.
- Network connection between source and destination is established.
- Required resources are reserved on the destination host.

2. Reservation

- CPU, memory, storage, and network resources are allocated on the destination host.
- Destination host prepares to receive the VM.

3. Iterative Pre-Copy Phase

- Memory pages of the running VM are copied to the destination.
- VM continues execution during copying.
- Modified (**dirty**) pages are tracked.
- Dirty pages are recopied in multiple rounds until only a few remain.

4. Stop-and-Copy Phase

- VM execution is temporarily suspended.
- Remaining dirty memory pages and CPU state are transferred.
- Causes only a very short downtime.

5. Commitment Phase

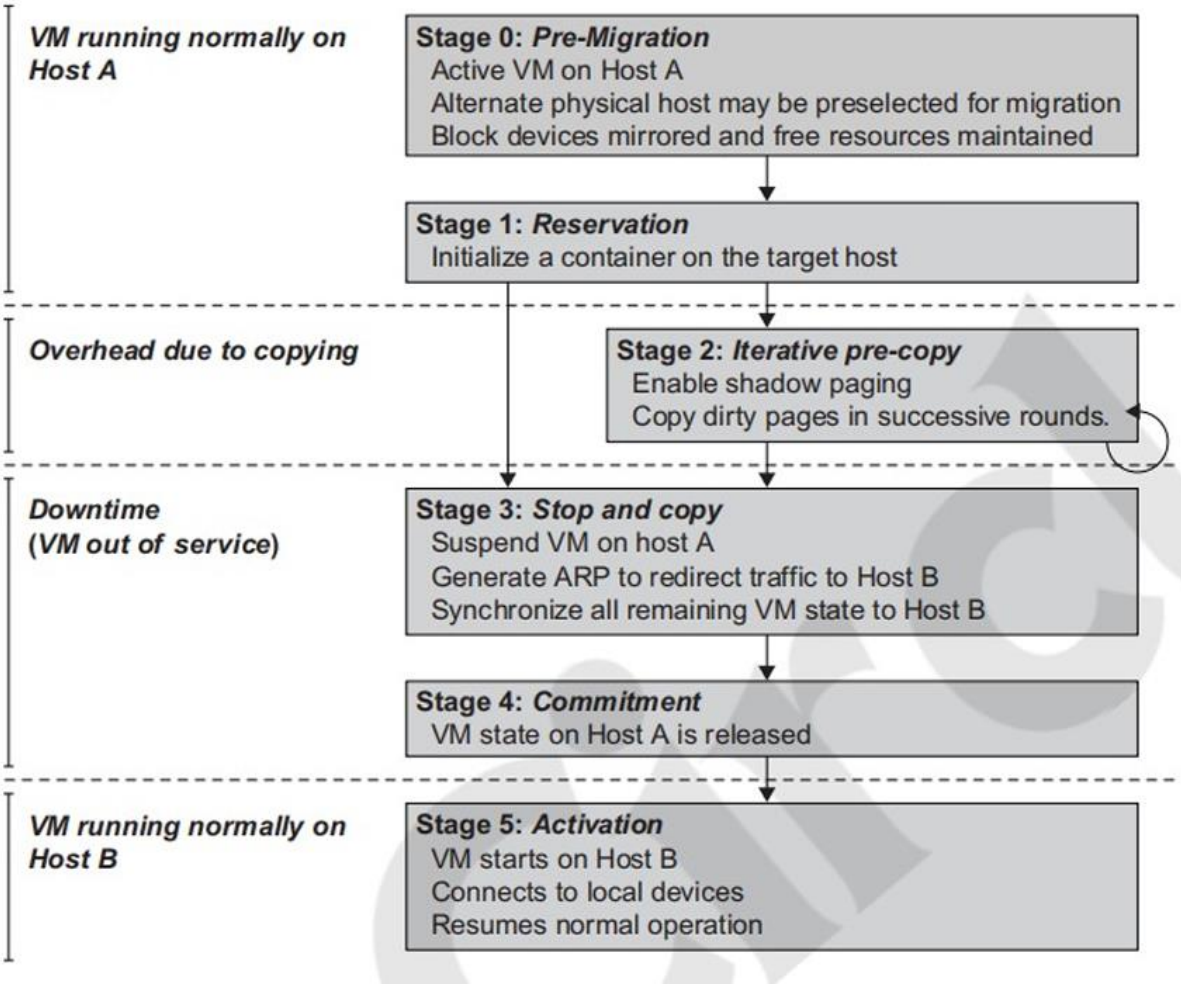
- Destination host confirms successful receipt of the VM state.
- Ownership of the VM is transferred.
- Source host releases allocated resources.

6. Activation Phase

- VM resumes execution on the destination host.
- Network connections are re-established.
- Users continue accessing services with minimal interruption.

Advantages of Live Migration

1. **Minimal service downtime.**
2. **Supports load balancing.**
3. **Enables server maintenance without shutting down VMs.**
4. **Improves fault tolerance and availability.**
5. **Optimizes resource utilization.**
6. **Supports energy-efficient data center operation.**



4b. Develop a architecture of livewire for Intrusion detection using a dedicated VM.

Livewire Intrusion Detection System (IDS)

Livewire is an Intrusion Detection System (IDS) designed for **virtualized environments**. It uses a dedicated **IDS Virtual Machine (IDS VM)** to monitor other virtual machines externally, making it difficult for attackers to detect or disable the monitoring system.

Components of Livewire Architecture

1. Physical Hardware

- Consists of:
  - CPU
  - Memory
  - Storage
  - Network resources
- Provides resources for all virtual machines.

2. Virtual Machine Monitor (VMM) / Hypervisor

- Runs directly on hardware.
- Manages virtual machines.
- Provides isolation between VMs.

3. Guest Virtual Machines

- Run user applications and operating systems.
- May become targets of attacks.
- Their activities are monitored by the IDS VM.

4. Dedicated IDS VM

- Separate virtual machine used exclusively for intrusion detection.
- Runs intrusion detection tools.
- Monitors guest VMs externally.
- Cannot be easily compromised by attacks inside guest VMs.

5. Livewire Monitoring Layer

- Collects information from guest VMs through the hypervisor.
- Monitors:
  - Memory
  - Processes
  - Files
  - Network activities
- Detects malicious behavior and intrusions.

Working of Livewire

1. Monitoring

- Dedicated IDS VM continuously monitors guest VMs.
- Collects system and network information.

2. Analysis

- IDS VM analyzes VM states and activities.
- Detects abnormal behavior and intrusion attempts.

3. Isolation

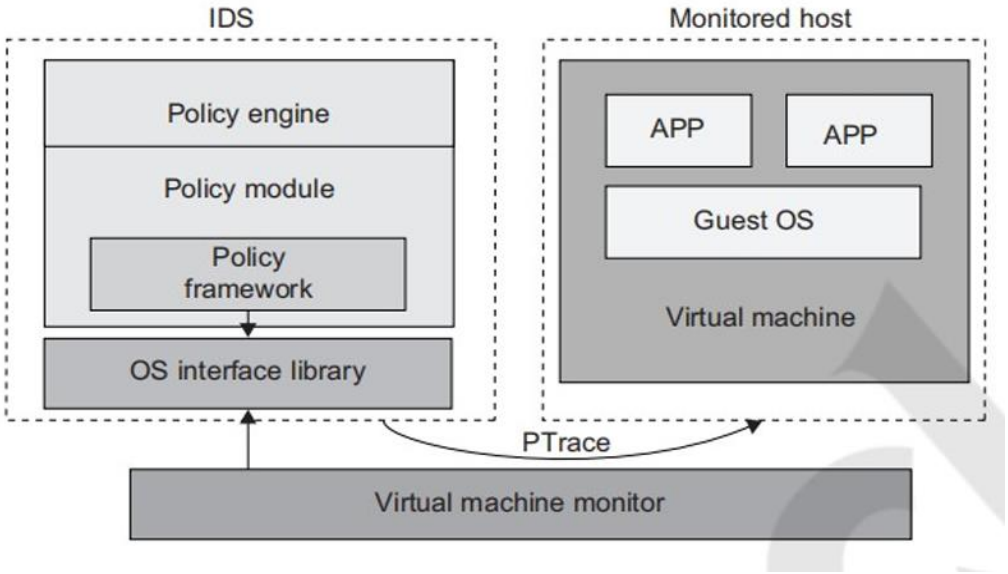
- Monitoring occurs outside the monitored VM.
- Attackers inside guest VMs cannot easily disable the IDS.

4. Detection and Alert

- Suspicious activities are identified.
- Alerts are generated for administrators.
- Appropriate security actions can be taken.

Advantages of Livewire

1. Strong isolation between monitoring system and guest VMs.
2. Difficult for attackers to detect or disable the IDS.
3. Improves security in virtualized environments.
4. Monitors memory, processes, files, and network traffic.
5. Supports early detection of attacks and intrusions.
6. Centralized monitoring of multiple virtual machines.



5a. Six Design Objectives for Cloud Computing

Cloud architecture must satisfy several requirements to provide efficient, scalable, secure, and reliable services.

1. Scalability

- The cloud should easily expand by adding more servers and resources.
- It must support increasing workloads without performance degradation.
- Allows organizations to handle growth efficiently.

Benefits

- Supports large numbers of users.
- Improves performance under heavy loads.

2. Virtualization

- Computing, storage, and networking resources are virtualized.
- Multiple virtual machines can share the same physical hardware.
- Enables flexible and efficient resource allocation.

Benefits

- Better resource utilization.
- Reduced hardware costs.

3. Efficiency

- Resources should be utilized effectively.
- Dynamic provisioning reduces resource wastage.
- Maximizes performance while minimizing operational costs.

Benefits

- Higher utilization rates.
- Cost-effective operation.

4. Reliability

- Cloud services should remain available even when failures occur.
- Data and services are replicated across geographically distributed data centers.
- Fault tolerance mechanisms ensure continuous operation.

Benefits

- High availability.
- Improved disaster recovery.

5. Security

- Shared resources and user data must be protected.
- Access control, authentication, encryption, and monitoring are required.
- Security remains one of the major challenges in cloud platforms.

Benefits

- Data confidentiality and integrity.
- Protection against cyberattacks.

6. Dynamic Resource Provisioning

- Resources should be automatically allocated and released according to demand.
- Supports elasticity and efficient resource management.

Benefits

- Faster response to workload changes.
- Reduced operational costs.

5b. With a neat diagram,build a cloud ecosystem with a private cloud

A **Private Cloud Ecosystem** is a cloud infrastructure owned and managed by a single organization. It provides cloud services with greater security, control, and flexibility.

Components of the Private Cloud Ecosystem

1. Cloud Consumers

- Individual users and other clouds act as consumers.
- They request cloud services and computing resources.
- Demand a flexible and scalable computing platform.

Functions

- Access cloud applications and services.
- Consume computing, storage, and network resources.

2. Cloud Management Layer

- Provides virtualized resources through an **Infrastructure as a Service (IaaS)** platform.
- Manages cloud services and resource provisioning.

Functions

- Resource allocation
- Service management
- Monitoring and control

3. Virtual Infrastructure (VI) Management

- Allocates and manages virtual machines across multiple server clusters.
- Handles VM deployment and scheduling of resources.

Functions

- VM placement
- Resource scheduling
- Load balancing

4. VM Management Layer

- Manages virtual machines running on individual hosts.
- Supports virtualization technologies such as:
    - Xen
    - KVM
    - VMware

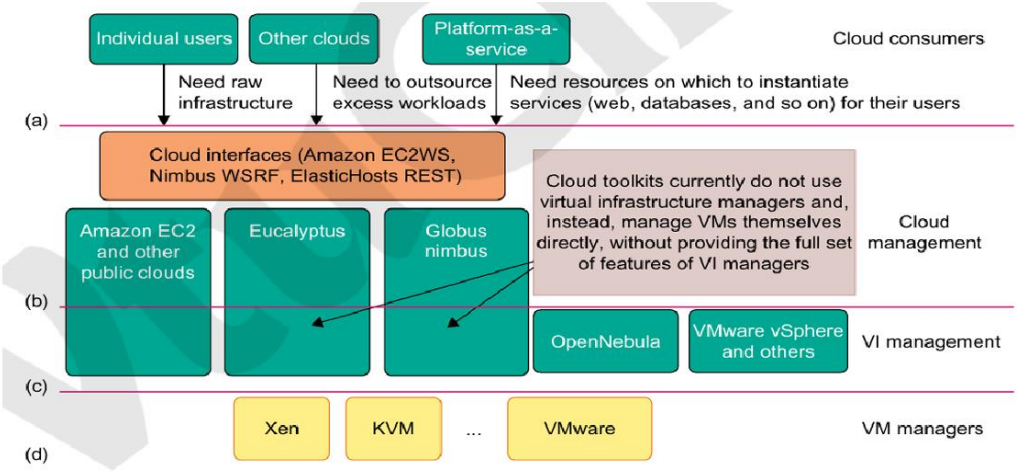
Functions

- VM creation and deletion

- VM monitoring
- VM migration and maintenance

Need for a Private Cloud Ecosystem

1. Enables organizations to build and manage their own cloud infrastructure.
2. Improves resource utilization through virtualization.
3. Provides flexibility and scalability.
4. Supports workload outsourcing to public clouds when required.
5. Offers better security, privacy, and administrative control.



5c. Organize Functional Modules of GAE

Google App Engine (GAE) is a **Platform as a Service (PaaS)** that provides an environment for developing, deploying, and managing cloud applications.

1. Runtime Environment

- Executes user applications on Google's infrastructure.
- Supports programming languages such as:
  - Python
  - Java
- Provides automatic scaling and load balancing.

Functions

- Application execution
- Resource management
- Automatic scalability

2. Software Development Kit (SDK)

- Provides tools for building, testing, and debugging applications locally.
- Enables developers to deploy applications to Google Cloud.

Functions

- Local development
- Testing and debugging
- Application deployment

3. Datastore Module

- Provides scalable data storage services.
- Based on Google's **BigTable** technology.
- Stores structured and semi-structured data.

Functions

- Data storage
- Data retrieval
- Scalability for large datasets

4. Services Module

- Provides commonly used services required by applications.

Services Include

- Authentication
- Email service
- Caching
- URL Fetch
- Image processing

Benefits

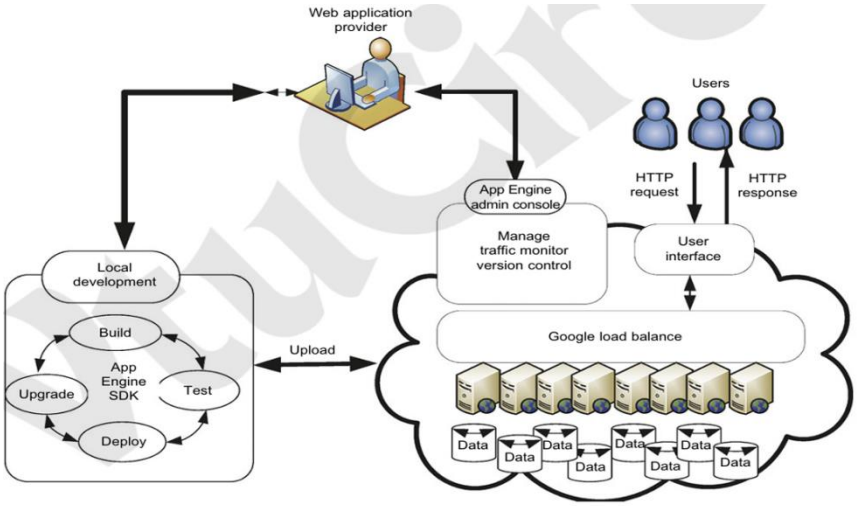
- Reduces development effort
- Improves application functionality

5. Web Interface Module

- Provides user access through web browsers.
- Handles HTTP requests and responses.
- Acts as the front-end interface between users and cloud applications.

Functions

- User interaction
- Request processing
- Response delivery



6a. Basic Requirements for Managing the Resources of a Data Center

Efficient data-center resource management is essential for providing cloud services with **high performance, reliability, scalability, and cost effectiveness**.

1. Resource Provisioning

- Allocate CPU, memory, storage, and bandwidth according to user demand.
- Avoid under-provisioning and over-provisioning of resources.

Benefits

- Efficient resource utilization
- Improved service performance

2. Service Level Agreement (SLA) Support

- Resources must satisfy Quality of Service (QoS) requirements specified in SLAs.
- Prevent SLA violations and penalties.

Benefits

- Customer satisfaction
- Guaranteed service quality

3. Dynamic Resource Allocation

- Resources are allocated and released automatically based on workload changes.
- Supports cloud elasticity.

Benefits

- Better scalability
- Reduced resource wastage

4. Virtual Machine Management

- Efficient VM deployment, migration, monitoring, and recovery.
- Supports virtualization-based cloud operation.

Benefits

- Flexible resource management
- Improved system reliability

5. Load Balancing

- Workloads should be distributed evenly among servers.
- Prevents server overload.

Benefits

- Improved utilization
- Better performance

6. Fault Tolerance and Reliability

- System should continue operation despite failures.
- Fast recovery and VM migration mechanisms are required.

Benefits

- High availability
- Reduced downtime

7. Monitoring and Accounting

- Resource usage and performance must be monitored continuously.
- Accounting mechanisms track resource consumption.

Benefits

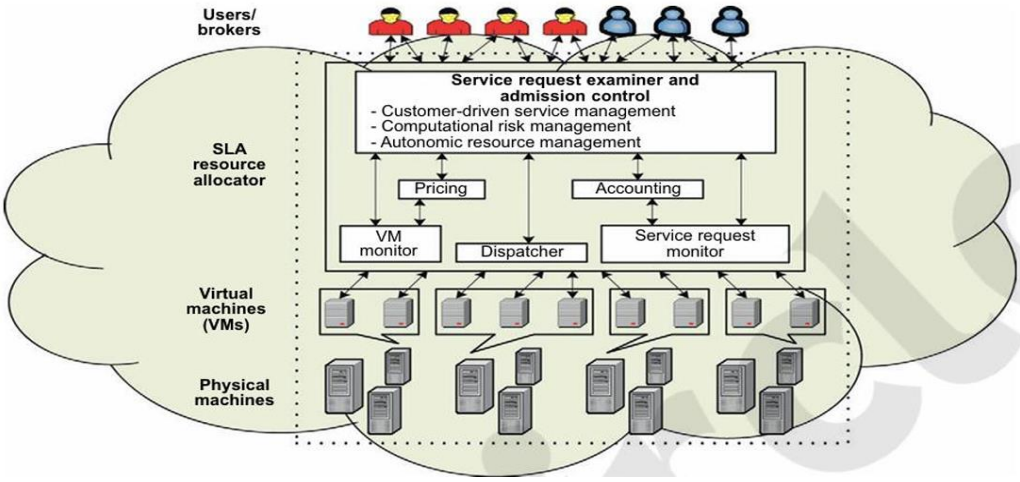
- Efficient management
- Accurate billing

8. Cost and Energy Efficiency

- Reduce operational costs and power consumption.
- Improve overall data-center utilization.

Benefits

- Lower expenses
- Environment-friendly operation



6b. Summarize Six Open Challenges in Cloud Architecture Development

1. Service Availability and Data Lock-In Problem

- Cloud services managed by a single provider may become a **single point of failure**.
- Data and applications cannot be easily moved between cloud providers due to proprietary APIs.
- Standardized APIs are required to improve portability and interoperability.

Challenges

- Vendor lock-in
- Service outages
- Limited portability

2. Data Privacy and Security Concerns

- Clouds store large amounts of sensitive and confidential data.
- Public cloud environments are vulnerable to:
  - DoS attacks
  - Malware
  - Spyware
  - VM hijacking
  - Hypervisor attacks
- Strong security mechanisms are necessary.

Solutions

- Encryption
- Firewalls
- Authentication and access control

3. Scalable Data Management and Storage

- Cloud systems must manage huge volumes of distributed data.
- Efficient mechanisms are required for:

    - Storage allocation
    - Data retrieval
    - Replication
    - Consistency management
- Data centers must support large-scale storage systems.

4. Service Scalability and Resource Provisioning

- User demand changes dynamically.
- Resources such as CPU, memory, storage, and bandwidth must be provisioned automatically.
- Both under-provisioning and over-provisioning should be avoided.

Requirements

- Elastic scaling
- Dynamic resource allocation
- Efficient workload management

5. Energy Efficiency and Power Management

- Large cloud data centers consume enormous amounts of electrical power.
- Efficient cooling and power management techniques are required.
- Energy-aware resource allocation reduces operational costs.

Benefits

- Lower energy consumption
- Reduced carbon footprint
- Cost savings

6. Standardization and Interoperability

- Different cloud providers use different interfaces and protocols.
- Lack of standards makes integration between clouds difficult.
- Universal APIs and access protocols are needed for cloud federation and portability.

Benefits

- Better interoperability
- Easier migration between providers
- Multi-cloud support

6c. Amazon Web Services

Amazon Web Services (AWS) is a cloud computing platform provided by **Amazon**. It offers a collection of web services that enable users to access computing, storage, networking, databases, messaging, monitoring, and other resources on a **pay-as-you-go** basis.

AWS is primarily an **Infrastructure as a Service (IaaS)** platform.

Features of AWS

- Provides **Elastic Compute Cloud (EC2)** for computing services.
- Provides **Simple Storage Service (S3)** for storage.
- Supports **Elastic MapReduce (EMR)** for large-scale data processing.
- Offers database services such as **SimpleDB** and **RDS**.
- Provides networking services through **Virtual Private Cloud (VPC)**.
- Supports **Auto Scaling** and **Load Balancing**.
- Uses a **pay-as-you-go** pricing model.

AWS Service Offerings

1. Compute Services

- **EC2 (Elastic Compute Cloud)** – Virtual servers for computing.
- **Elastic MapReduce (EMR)** – Large-scale data processing.
- **Auto Scaling** – Automatically adjusts resources according to workload.

2. Storage Services

- **S3 (Simple Storage Service)** – Object storage service.
- **EBS (Elastic Block Store)** – Persistent block storage for EC2 instances.

3. Database Services

- **SimpleDB** – Non-relational cloud database.
- **RDS (Relational Database Service)** – Managed relational database service.

4. Messaging Services

- **SQS (Simple Queue Service)** – Message queuing service.
- **SNS (Simple Notification Service)** – Notification and messaging service.

5. Networking Services

- **VPC (Virtual Private Cloud)** – Secure virtual network environment.
- **Elastic Load Balancing (ELB)** – Distributes traffic across multiple servers.

6. Monitoring Services

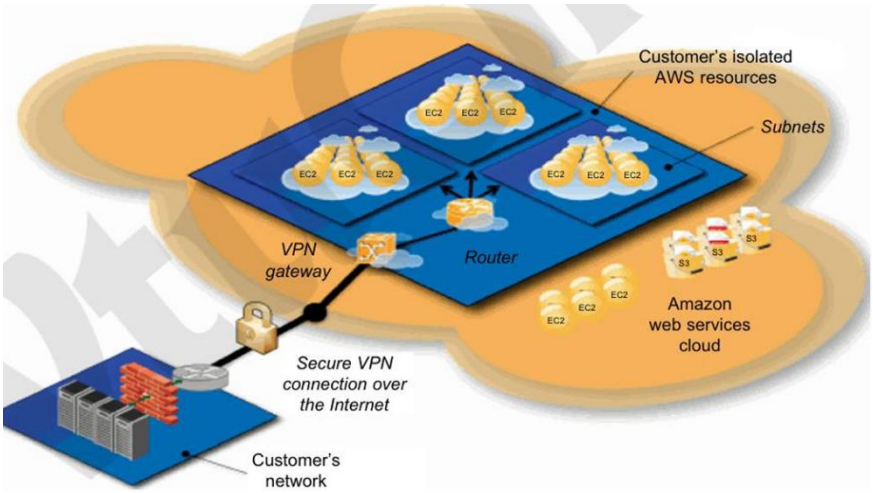
- **CloudWatch** – Monitors AWS resources and applications.

7. Content Delivery Service

- **CloudFront** – Content Delivery Network (CDN) for faster content distribution.

8. Payments and Billing Services

- **FPS (Flexible Payments Service)** – Online payment processing.
- **Amazon DevPay** – Billing and payment management for developers.



7a. Demonstrate surfaces of attacks in a cloud computing environment with neat diagram

In cloud computing, attackers can target different components of the virtualized environment. The major attack surfaces include the **Hypervisor, Virtual Machines, Virtual Networks, Virtual Storage, and VM Images**.

. Hypervisor Attack Surface

- The **Hypervisor (VMM)** manages all virtual machines.
- If compromised, attackers can gain control over multiple VMs.
- Considered one of the most dangerous cloud security threats.

Risks

- Complete system compromise
- Access to hosted virtual machines
- Loss of isolation

2. Virtual Machine (VM) Attack Surface

- Guest VMs may contain software vulnerabilities.
- Attackers can exploit these vulnerabilities to gain unauthorized access.

Risks

- VM compromise
- Data theft
- Integrity violations

3. VM Escape Attacks

- An attacker escapes from a guest VM to the hypervisor.
- May gain access to other VMs running on the same physical host.

Risks

- Cross-VM attacks
- Hypervisor compromise
- Unauthorized access

4. Rogue Virtual Machines

- Unauthorized or malicious VMs are created within the cloud environment.
- These VMs may consume resources or attack other VMs.

Risks

- Resource misuse
- Malware distribution
- Internal attacks

5. Virtual Network Attack Surface

- Virtual networks can be targeted just like physical networks.
- Data exchanged between VMs may be intercepted or modified.

Risks

- Eavesdropping
- Packet sniffing
- Man-in-the-middle attacks

6. Virtual Storage and VM Images

- Insecure VM images may contain malware or vulnerabilities.
- Sensitive information stored in virtual storage may be exposed.

Risks

- Data leakage
- Malware propagation
- Unauthorized access to stored data

7. Side-Channel Attacks

- Attackers exploit shared physical resources such as CPU cache, memory, or processors.
- Information about other VMs may be inferred.

Risks

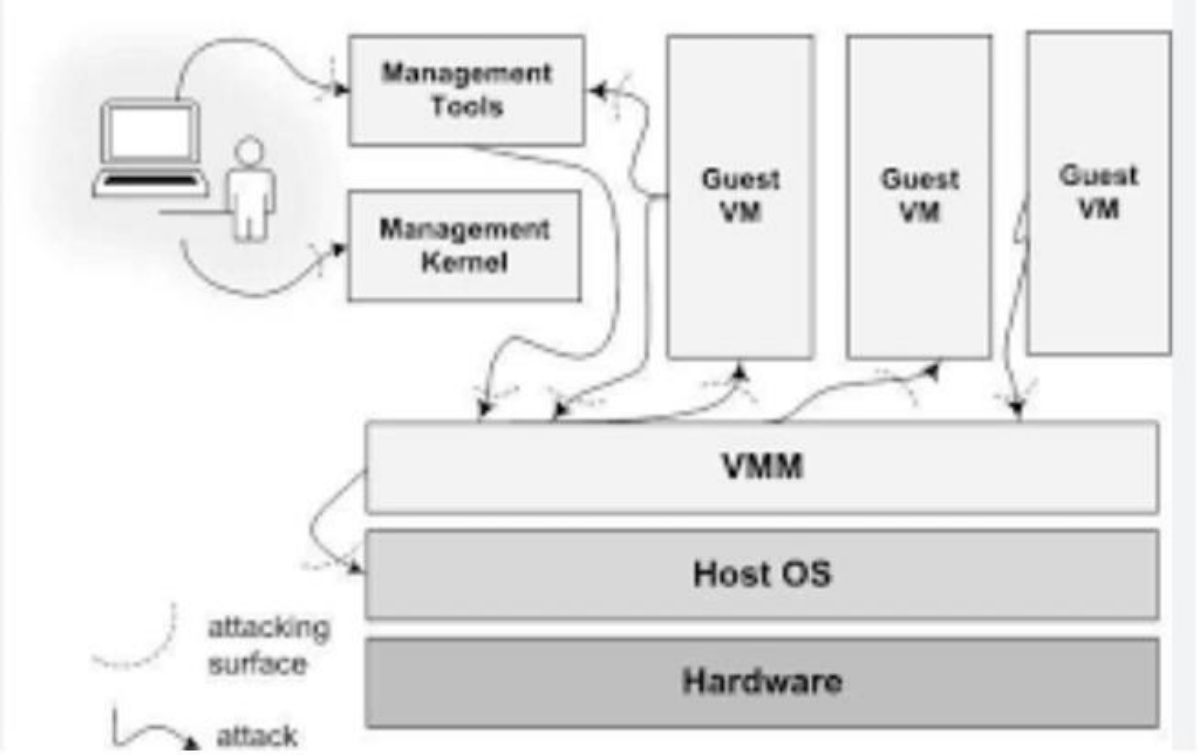
- Confidentiality breaches
- Information leakage

8. Resource Starvation Attacks

- Malicious VMs consume excessive CPU, memory, storage, or bandwidth.
- Legitimate users receive fewer resources.

Risks

- Reduced performance
- Service degradation
- Denial of Service (DoS)



7b. List out the top cloud security threats of CSA2016.

According to the **Cloud Security Alliance (CSA) 2016 Report**, the **Top Twelve Cloud Security Threats** are:

- **Data Breaches**
  - Loss of financial, health, and proprietary data.
- **Compromised Credentials and Weak Authentication**
  - Caused by poor password security and weak authentication mechanisms.

- **Hacked APIs**
  - Weak application interfaces expose cloud systems to attackers.
- **Exploited System Vulnerabilities**
  - Attackers exploit known software and system flaws.
- **Account Hijacking**
  - Cybercriminals steal cloud user accounts and credentials.
- **Malicious Insiders**
  - Employees or trusted users misuse their privileged access.
- **Advanced Persistent Threats (APTs)**
  - Long-term, hidden cyberattacks used for espionage.
- **Permanent Data Loss**
  - Data becomes unrecoverable due to accidental deletion or insufficient backups.
- **Inadequate Diligence**
  - Failure to properly assess risks before adopting cloud services.
- **Cloud Service Abuse**
  - Cloud resources are misused for botnets, malware hosting, or spamming.
- **Denial of Service (DoS) Attacks**
  - Attackers overwhelm cloud services, making them unavailable.
- **Shared Technology Risks**
  - Multi-tenancy vulnerabilities in hypervisors and virtual machines.

7c. Select four widely-accepted fair information practices that "Consumer oriented commercial web sites that collect personal identifying information from or about consumers online would be required to comply with.

Fair Information Practices (FIPs) are guidelines developed to protect the privacy of consumers whose personal information is collected online. According to the **Federal Trade Commission (FTC)**, commercial websites collecting personal identifying information should follow these principles.

1. Notice (Awareness)

□

Websites must clearly inform users about their information practices.

Users should know:

- What information is being collected.
- Why it is being collected.
- How it will be used.
- Whether it will be shared with third parties.

Privacy policies should be easily accessible.

2. Choice (Consent)

Consumers should have the right to decide how their information is used.

Websites should provide options to:

- Accept or reject data collection.
- Allow or deny sharing of personal information with others.

Gives users control over their personal data.

3. Access (Participation)

Consumers should be able to view the personal information stored about them.

They should have the ability to:

- Correct inaccurate information.
- Update outdated information.
- Request modifications when necessary.

Ensures data accuracy and fairness.

4. Security

Organizations must protect personal information from:

- o Unauthorized access.
- o Modification.
- o Disclosure.
- o Destruction.

Security measures such as encryption, authentication, and access control should be implemented

8a. Summarize The design goals of Xoar are:

Xoar is a modified version of Xen that follows microkernel principles to improve security by reducing the attack surface and minimizing the Trusted Computing Base (TCB).

Design Goals of Xoar

1. Maintain the Functionality of Xen

- Xoar should provide the same core functionality as Xen.
- Security improvements should not affect normal virtualization operations.

2. Ensure Transparency

- Existing management tools and VM interfaces should continue to work.
- Users should not need major changes to use Xoar.

3. Tightly Control Privileges

- Each component should have only the minimum privileges required.
- This follows the principle of least privilege and reduces security risks.

4. Minimize Component Interfaces

- Reduce communication interfaces between components.
- Fewer interfaces mean fewer attack vectors for attackers.

5. Eliminate Unnecessary Sharing

- Components should share only essential resources.
- This reduces the possibility of unauthorized access and information leakage.

6. Explicitly Define Sharing

- All sharing between components should be clearly defined.
- Enables proper logging, monitoring, and auditing of activities.

7. Reduce Attack Exposure Window

- Components should remain active only when needed.
- Minimizes the time available for attackers to exploit vulnerabilities.

8b. Explain mobile devices and cloud security

Mobile Device Security Issues

1. Loss or Theft of Devices

- Mobile devices can be lost or stolen easily.
- Attackers may gain access to sensitive cloud data stored on the device.

2. Malware Attacks

- Smartphones and tablets are vulnerable to viruses, worms, Trojans, and spyware.
- Malware can steal passwords and personal information.

3. Insecure Wireless Networks

- Public Wi-Fi networks may allow attackers to intercept data.
- This can lead to unauthorized access to cloud services.

4. Weak Authentication

- Weak passwords increase the risk of account compromise.
- Strong authentication mechanisms are required.

Cloud Security for Mobile Devices

1. Data Encryption

- Data should be encrypted during storage and transmission.
- Encryption protects sensitive information from attackers.

2. Access Control and Authentication

- Multi-factor authentication and access control should be implemented.
- Only authorized users should access cloud resources.

3. Secure Data Storage

- Cloud providers should ensure secure storage and backup of data.
- Proper security policies help prevent data leakage.

4. Continuous Monitoring

- Cloud systems should monitor suspicious activities and threats.
- Regular security updates improve protection.

8c. Model an overview of reputation system design options

A **reputation system** is used in cloud computing to establish trust among cloud users and service providers.It collects feedback, evaluates behavior, and assigns reputation scores to help users make informed decisions about cloud services. This is part of **Reputation-Guided Protection of Data Centers** in cloud security and trust management.

Reputation System Design Options

1. Centralized Reputation System

- A central authority collects and maintains reputation information.
- All users depend on a single repository for reputation scores.
- Easy to manage but creates a single point of failure.

2. Distributed Reputation System

- Reputation information is maintained across multiple nodes.
- No single authority controls the reputation data.
- More scalable and fault tolerant than centralized systems.

3. Feedback-Based Reputation System

- Reputation is calculated using ratings and feedback from users.
- Positive feedback increases reputation, while negative feedback decreases it.
- Helps identify trustworthy service providers.

4. Trust-Based Reputation System

- Uses trust relationships among users and cloud entities.
- Recommendations from trusted entities carry more weight.
- Improves the accuracy of reputation evaluation.

5. Global Reputation System

- A single reputation score is maintained for each cloud service.
- Allows easy comparison between different providers.
- Simple and easy to understand.

6. Personalized Reputation System

- Reputation depends on the user's own preferences and experiences.

- Different users may assign different trust values to the same service.
- Provides customized trust assessment.

7. Hybrid Reputation System

- Combines multiple reputation approaches.
- Uses both feedback and trust relationships.
- Provides higher reliability and better decision making

9a. Outline Important Cloud Platform Capabilities

Cloud platforms provide a set of capabilities that support the development, deployment, and execution of cloud applications. These capabilities enable **utility computing**, **elastic scaling**, **distributed storage**, and **large-scale data processing**. Cloud platforms hide infrastructure complexity and provide services to application developers.

Cloud Platform Capabilities

1. Physical or Virtual Computing Platform

- Provides physical or virtual machines for executing applications.
- Virtualization enables efficient resource sharing and isolation.
- Forms the basic computing infrastructure of the cloud.

2. Massive Data Storage Service and Distributed File System

- Provides large-scale storage facilities.
- Supports distributed file systems for storing huge amounts of data.
- Offers file access similar to local file systems.

3. Massive Database Storage Service

- Provides scalable database services.
- Helps store and manage structured data efficiently.
- Similar to traditional DBMS but designed for cloud-scale applications.

4. Massive Data Processing and Programming Models

- Supports large-scale data processing on thousands of nodes.
- Provides programming models such as MapReduce.
- Handles failures and scalability automatically.

5. Workflow and Query Language Support

- Provides workflow systems for application execution.
- Supports data query languages for data access and management.
- Simplifies complex cloud application development.

6. Programming Interface and Service Deployment

- Provides APIs and web interfaces for cloud services.
- Supports technologies such as J2EE, PHP, ASP, and Rails.

- Facilitates easy deployment of cloud applications.

7. Runtime Support

- Includes distributed monitoring services.
- Provides distributed task scheduling.
- Supports distributed locking mechanisms.
- Runtime services are transparent to users.

8. Support Services

- Offers data and computing services.
- Supports data-parallel programming models such as MapReduce.
- Helps execute large-scale distributed applications efficiently.

9b. Organize the steps involved in MapReduce

MapReduce is a programming model developed by Google for processing and generating large datasets on clusters of computers. It simplifies distributed computing by dividing the computation into **Map** and **Reduce** operations while automatically handling scheduling, data distribution, synchronization, and fault tolerance.

Steps Involved in MapReduce

Step 1: Input Data Splitting

- The input file is divided into several fixed-size splits.

- Each split is assigned to a Map worker.

- This enables parallel processing of large datasets.

Step 2: Map Phase

- The user-defined **Map()** function processes each input split.

- It reads input records and generates intermediate **(key, value)** pairs.

- Multiple Map tasks execute simultaneously on different nodes.

Step 3: Intermediate Data Storage

- The intermediate key-value pairs produced by Map tasks are stored locally.

- The master keeps track of the locations of these intermediate results.

Step 4: Partitioning

- Intermediate data is partitioned into regions.

- Each partition is assigned to a specific Reduce task.

- Typically done using a partitioning function based on the key.

Step 5: Shuffle Phase

- Reduce workers fetch intermediate data from Map workers.

- All values associated with the same key are transferred to the same Reduce worker.

- This process is known as **Shuffle**.

Step 6: Sort Phase

- Intermediate keys are sorted before reduction.

- Values with identical keys are grouped together.

- Sorting makes aggregation efficient.

Step 7: Reduce Phase

- The **Reduce()** function processes each key and its associated values.

- It combines, summarizes, or aggregates the results.

- Produces the final output.

Step 8: Output Generation

- Final results are written to output files.

- Output is stored in the distributed file system.

- Multiple output files may be generated, one per Reduce task.

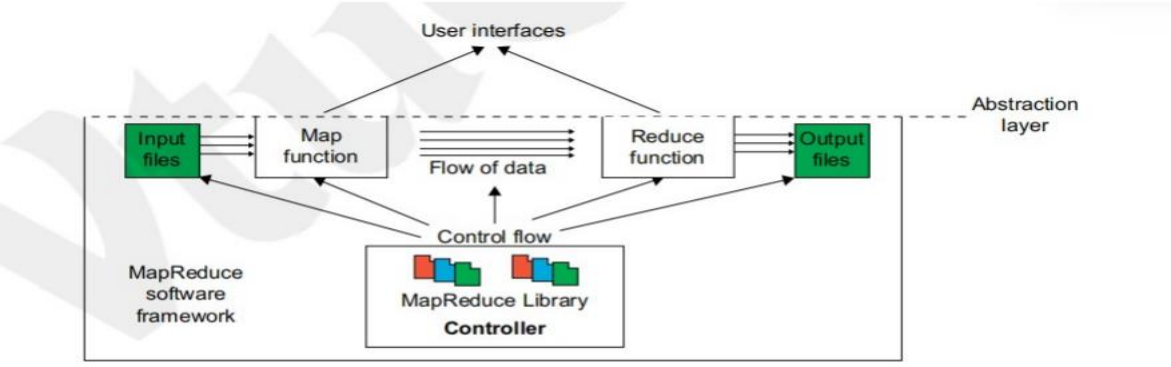


FIGURE 6.1

MapReduce framework: Input data flows through the Map and Reduce functions to generate the output result under the control flow using MapReduce software library. Special user interfaces are used to access the Map and Reduce resources.

10a. Explain with a neat diagram how data flows in running a MapReduce job at various task trackers using the Hadoop library.

Hadoop is an open-source implementation of MapReduce. The Hadoop MapReduce engine runs on top of HDFS and follows a **Master–Worker architecture**. The data flow of a MapReduce job involves **Map tasks running on Task Trackers (workers)** and **Reduce tasks collecting and processing intermediate results**.

Data Flow of a MapReduce Job

1. Data Partitioning

- The input data stored in HDFS is divided into **M input splits**.
- Each split corresponds to one Map task.

- The Hadoop library automatically performs this partitioning.

2. Creation of Master and Workers

- One process acts as the **Master (JobTracker)**.
- Remaining processes become **Workers (TaskTrackers)**.

- The master assigns Map and Reduce tasks to idle workers.

3. Reading Input Data

- Each TaskTracker running a Map task reads its assigned input split from HDFS.
- Data is supplied to the Map function as **(key, value)** pairs.

4. Map Phase

- The user-defined **Map()** function processes each input pair.
- It generates intermediate **(key, value)** pairs.
- Example: Word Count generates pairs like:

  (people,1)

  (most,1)

  (poetry,1)

5. Combiner Phase (Optional)

- A local Combiner may run on each TaskTracker.
- It performs partial aggregation of intermediate results.
- This reduces network traffic during data transfer.

6. Partitioning

- Intermediate data is divided into **R partitions**, where R is the number of Reduce tasks.
- A partitioning function such as:

  Hash(Key) mod R

ensures that identical keys go to the same reducer.

7. Shuffle and Synchronization

- After all Map tasks finish, reducers begin fetching their partitions.
- Reduce workers collect intermediate data from all Map workers.
- This stage is called **Shuffle**.

8. Sorting and Grouping

- Each reducer sorts the received data by key.
- Values with identical keys are grouped together.

9. Reduce Phase

- The **Reduce()** function processes each group.
- It combines values belonging to the same key.

10. Writing Output

Final results are written into output files.

Output files are stored back in **HDFS**

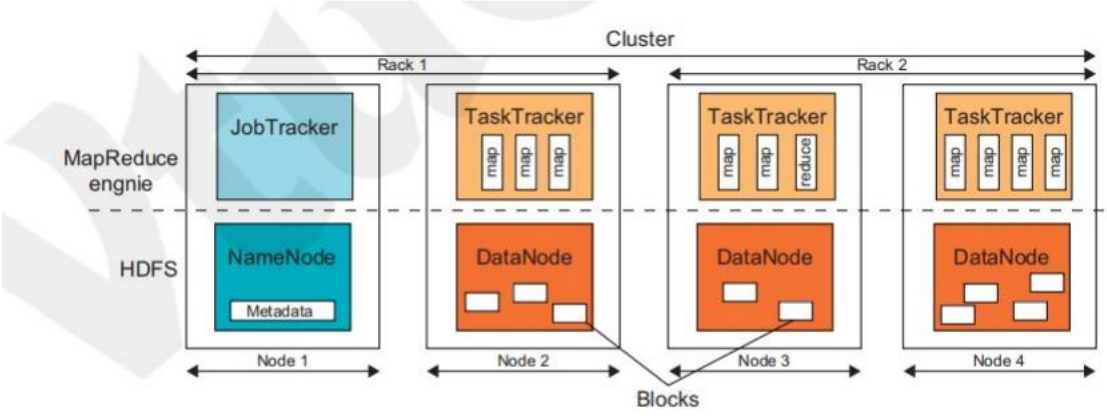


FIGURE 6.11

HDFS and MapReduce architecture in Hadoop where boxes with different shadings refer to functional nodes applied to different blocks of data.

10b. Explain Data mutation sequence in GFS with diagram

Data mutation in Google File System (GFS) refers to operations that modify file contents such as **write** and **append** operations. GFS uses a **primary replica** and multiple **secondary replicas** to maintain consistency among replicated chunks.

Data Mutation Sequence in GFS

Step 1: Client Contacts Master

- The client asks the **master** which chunk server currently holds the **lease (primary replica)** for the chunk.
- It also requests the locations of other replicas.

Step 2: Master Replies

- The master returns:
  - Identity of the **primary replica**
  - Locations of **secondary replicas**
- The client caches this information for future use.

Step 3: Client Pushes Data

- The client pushes data to **all replicas**.
- Data can be sent in any order.
- Each chunk server stores the data temporarily in an internal buffer cache.
- Data flow is separated from control flow to improve performance.

Step 4: Write Request to Primary

- After all replicas acknowledge receipt of data, the client sends a **write request** to the primary

- replica.
- The request identifies the data previously pushed.
- The primary assigns a **serial number** to maintain proper ordering of mutations.

Step 5: Forward Request to Secondaries

- The primary forwards the write request to all secondary replicas.
- Each secondary applies the mutation in the same serial order assigned by the primary.

Step 6: Secondary Acknowledgement

- All secondary replicas complete the operation.
- They send acknowledgements back to the primary replica.

Step 7: Primary Replies to Client

- The primary confirms successful completion to the client.
- If any error occurs, it is reported to the client.
- The client retries the mutation if required.

Error Handling in GFS

- If a failure occurs, the write may succeed on the primary and only some secondary replicas.
- This may leave replicas temporarily inconsistent.
- The client retries the mutation operation until consistency is restored.

Advantages of GFS Data Mutation

1. High consistency among replicas.
2. Efficient write and append operations.

3. Fault tolerance through replication.4. High availability and scalability.

5. Reduced master involvement after lease assignment.

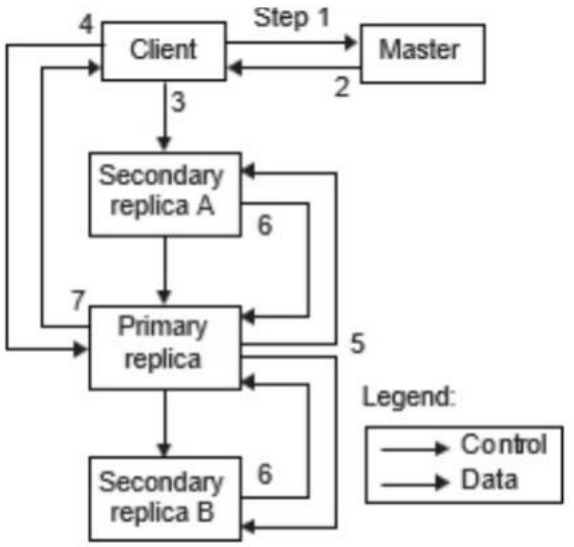


FIGURE 6.19

Data mutation sequence in GFS.