1 a. Why is cloud computing considered an evolution rather than as an innovation? What are the advantages of grid computing?

➢ Cloud Computing as an Evolution Rather than an Innovation (4 Marks)

Cloud computing is considered an **evolution** rather than a **pure innovation** because it builds upon existing technologies such as distributed computing, virtualization, and the internet. It enhances and refines these technologies rather than introducing something entirely new. The key reasons are:

1. **Foundation on Existing Technologies** – Cloud computing evolved from earlier concepts like mainframe computing, grid computing, and utility computing.
2. **Advancement of Virtualization** – Virtualization, a core aspect of cloud computing, existed before but was improved for scalable cloud solutions.
3. **Improved Resource Utilization** – Cloud computing refines resource-sharing models from grid and cluster computing.
4. **Internet & Web Services Expansion** – The rise of the internet and webbased services accelerated the shift to cloud-based solutions.

Thus, cloud computing is an **advancement of pre-existing models** rather than a completely new technological breakthrough.

Advantages of Grid Computing (3 Marks)

Grid computing is a form of distributed computing where multiple computers work together to solve complex problems. Its advantages include:

1. **High Computational Power** – Utilizes multiple systems to perform tasks that require large processing power.
2. **Cost Efficiency** – Reduces the need for expensive supercomputers by using available networked resources.
3. **Resource Sharing** – Allows different organizations or users to share computing resources effectively.
4. **Fault Tolerance** – If one node fails, the task can be reassigned to another, ensuring reliability.
5. **Scalability** – Easily expands by adding more nodes, making it flexible for different workloads.

These benefits make grid computing valuable for scientific research, data analysis, and large-scale computations.

1b. Pokhara university has significant number of students and conducts various examinations throughout of the year. The university wants to deploy cloud computing for its exam Department. Create a scenario that outlines which deployment model can be used? [8]

➢ **Scenario: Cloud Computing Deployment Model for Pokhara University's Exam Department**

Pokhara University plans to integrate **cloud computing** to streamline its **examination department**, ensuring secure, scalable, and efficient management of exam-related activities. Given the university's requirements, the most suitable **deployment model** is the **Private Cloud**.

Justification for Private Cloud (4 Marks)

1. **Data Security & Confidentiality** – Exam-related data, including question papers, student records, and results, must be kept confidential. A **private cloud** ensures restricted access to authorized personnel only.
2. **Customizability & Control** – The university can tailor the cloud environment based on its needs, such as online examination platforms, result processing systems, and secure storage.
3. **Compliance with Regulations** – Universities must adhere to education and data protection policies. A private cloud allows better compliance with **government regulations** and **internal policies**.
4. **Dedicated Resources** – Since a private cloud is used exclusively by the university, there are no resource-sharing issues, ensuring **high performance and reliability** during peak exam times.

Cloud-Based Examination Scenario (4 Marks)

Pokhara University sets up a ***private cloud*** for its ***examination department***, where:

- **Question Paper Management** – Faculty securely upload and manage question papers in an **encrypted cloud repository**.
- **Online Exam Platform** – Students can access **secured online exams** through a **university-authenticated login.**

- **Automated Result Processing** – Cloud-based systems process exam results, reducing manual workload and errors.
- **Data Backup & Disaster Recovery** – Cloud storage ensures **automatic backups**, preventing data loss due to system failures.
- **Scalability for Exam Periods** – During peak exam seasons, computing power can be dynamically allocated to **handle high traffic**.

A **Private Cloud** offers **security, scalability, and control**, making it the ideal choice for Pokhara University's exam department.

2a. what is Infrastructure-as-a-Service? Explain the backend architecture of cloud computing. [7]

➢ Infrastructure-as-a-Service (IaaS) (3 Marks)

Infrastructure-as-a-Service (IaaS) is a cloud computing model that provides virtualized computing resources over the internet. It offers fundamental infrastructure such as virtual machines, storage, networking, and operating systems on a pay-as-you-go basis.

Key Features of IaaS:

Scalability – Users can scale computing power up or down as needed.

Cost-Efficiency – No need for physical hardware investment; resources are rented.

On-Demand Resources – Provides servers, storage, and networking on demand.

Flexibility – Users can install and manage their own software and applications.

Managed Infrastructure – The cloud provider manages hardware, while users control OS and applications.

Examples of IaaS Providers:

Amazon Web Services (AWS) – EC2

Microsoft Azure – Virtual Machines

Google Cloud – Compute Engine

Backend Architecture of Cloud Computing (4 Marks)

The backend architecture of cloud computing is responsible for processing and managing user requests. It includes:

Cloud Infrastructure (Hardware Layer) – Physical servers, storage devices, and network components in data centers.

Virtualization Layer – Uses hypervisors (like VMware, Xen, or KVM) to create and manage virtual machines (VMs).

Resource Management – Allocates and optimizes resources (CPU, RAM, storage) dynamically based on demand.

Security & Compliance – Implements firewalls, encryption, authentication mechanisms to ensure data protection.

APIs & Middleware – Provides interfaces for users and developers to interact with the cloud services.

Storage Management – Manages block storage, object storage, and database storage for efficient data handling.

This backend infrastructure ensures high availability, scalability, and security in cloud computing environments.

2b. why is Service Oriented Architecture considered as the emergence of flexible application architecture? [8]

- **Service-Oriented Architecture (SOA) as the Emergence of Flexible Application Architecture**

Service-Oriented Architecture (SOA) is considered the **emergence of flexible application architecture** because it enables **modular, reusable, and scalable** application development. It allows different software components (services) to communicate over a network, making applications more adaptable and efficient.

Key Reasons Why SOA Provides Flexibility (4 Marks)

Loose Coupling – Services operate independently, meaning changes in one service do not impact others, ensuring flexibility in development and maintenance.

Reusability – Common functionalities (e.g., authentication, payment processing) are developed once and reused across multiple applications, reducing redundancy.

Interoperability – SOA allows services to communicate across different platforms and programming languages using standardized protocols like **SOAP and REST**.

Scalability – Applications can scale easily by adding or modifying services without overhauling the entire system.

SOA and the Evolution of Application Architecture (4 Marks)

Before SOA: Applications were developed as **monolithic structures**, meaning all components were tightly integrated, making updates difficult.

With SOA: Applications are broken into **independent, reusable services**, which can be updated or replaced without affecting the entire system.

Foundation for Cloud & Microservices: SOA laid the groundwork for **cloud computing** and **microservices architecture**, allowing businesses to develop dynamic, cloud-based applications.

SOA revolutionized application architecture by promoting **modularity, reusability, and platform independence**, making applications more **adaptive and scalable** in modern IT environments.

3a. ShopEasy is a popular online retailer known for offering a wide range of products, from electronics to clothing. As the business grew, the monolithic architecture that powered its e-commerce platform faced several challenges. ShopEasy wants to transform its monolithic e-commerce platform into a scalable and agile system using a microservices architecture. Design a microservices architectural journey of ShopEasy to improve flexibility, scalability, and faster development cycles. [5+10]

➢ ShopEasy’s Micro services Architectural Journey

1. Challenges with Monolithic Architecture (5 Marks)

As ShopEasy’s business expanded, its monolithic architecture created bottlenecks:

Scalability Issues – A single application structure made it hard to scale individual services.

Slow Development & Deployment – Changes in one module required redeploying the entire application.

Reliability Risks – A failure in one part of the system could bring down the entire platform.

Technology Lock-in – Limited flexibility to adopt new technologies.

To overcome these challenges, ShopEasy decided to migrate to Micro services Architecture for improved scalability, agility, and resilience.

---

2. Micro services Architecture Design for Shop Easy (10 Marks)

Step 1: Breaking Down the Monolith into Micro services

Shop Easy's platform is decomposed into independent micro services based on business functionalities:

User Service – Manages customer registration, authentication, and profiles.

Product Catalog Service – Handles product listings, categories, and inventory.

Order Management Service – Processes orders, payments, and shipping.

Cart Service – Manages shopping cart operations.

Payment Service – Processes transactions securely.

Recommendation Service – Uses AI to suggest products to users.

Notification Service – Sends emails and SMS notifications for orders and offers.

Each micro service is developed, deployed, and scaled independently.

Step 2: Technology Stack Selection

Backend: Node.js (for API development), Python (for AI-based recommendations)

Databases: MySQL for orders & users, MongoDB for product catalog, Redis for caching

Communication: REST APIs & gRPC for internal service communication

Containerization: Docker & Kubernetes for deployment

API Gateway: Manages authentication, load balancing, and routing

---

Step 3: Deployment & Scaling Strategy

Containerization & Orchestration: Docker + Kubernetes for efficient deployment and scaling.

Load Balancing: Implemented using NGINX and API Gateway to distribute traffic.

Auto-Scaling: Kubernetes Horizontal Pod Auto scaler (HPA) scales services dynamically.

Database Sharding: Distributed databases for improved query performance.

Step 4: Continuous Integration & Deployment (CI/CD)

CI/CD Pipeline: Jenkins/GitHub Actions automates testing and deployment.

Service Monitoring: Prometheus & Grafana for real-time insights and alerts.

Shop Easy’s shift to Micro services Architecture enhances scalability, development speed, and fault isolation, making it future-proof for further growth and innovation.

4a. how can the capacity of physical IT resources be used to their potential? Explain threats of hypervisor used in virtualization. [5+10]

➢ Utilizing Physical IT Resources to Their Full Potential (5 Marks)

To maximize the capacity of physical IT resources, organizations use various optimization techniques, including:

Virtualization – Allows multiple virtual machines (VMs) to run on a single physical server, improving hardware utilization.

Load Balancing – Distributes workloads evenly across servers to prevent underutilization and overloading.

Resource Pooling – Combines computing, storage, and network resources dynamically based on demand.

Containerization – Uses lightweight containers (e.g., Docker, Kubernetes) to run multiple applications efficiently on a single machine.

Dynamic Resource Allocation – Implements auto scaling to allocate CPU, RAM, and storage based on real-time workload needs.

These techniques ensure higher efficiency, reduced costs, and better performance of IT resources.

Threats to Hypervisors in Virtualization (10 Marks)

A hypervisor is a critical component in virtualization that allows multiple VMs to run on a single physical machine. However, it introduces several security threats, including:

Hyper jacking – Attackers install a malicious hypervisor to gain complete control over the virtual machines.

VM Escape – A vulnerability that allows a compromised VM to break out and access the hypervisor or other VMs.

Denial of Service (DoS) Attacks – Overloading the hypervisor with requests can cause service failures and downtime.

Data Leakage – Unauthorized access to virtual machine memory or storage can expose sensitive data.

Side-Channel Attacks – Attackers analyze shared resource usage (CPU cache, network traffic) to extract confidential information.

Weak Authentication & Privilege Escalation – Poorly configured access controls can allow unauthorized users to gain administrative privileges.

Rootkit Infections – Attackers may install persistent malware at the hypervisor level, making it difficult to detect and remove.

Insider Threats – Employees with high-level access can exploit hypervisor vulnerabilities for malicious purposes.

Insecure VM Migration – If VM data is not properly encrypted during migration, it can be intercepted by attackers.

Firmware & Patch Vulnerabilities – Unpatched hypervisors can be exploited using known security flaws.

Mitigation Strategies:

Implement strong access controls & multi-factor authentication (MFA).

Use hypervisor security tools to detect abnormal activities.

Regularly update and patch the hypervisor software.

Isolate critical VMs from less secure environments.

Encrypt VM migrations to prevent data interception.

By addressing these threats, organizations can secure their virtualization environments and ensure safe IT operations.

5 a. Explain different types of cloud security threats. [7]

Types of Cloud Security Threats (7 Marks)

Cloud computing introduces several security risks that organizations must address to protect data, applications, and services. The major cloud security threats include:

1. Data Breaches (1 Mark)

Unauthorized access to sensitive cloud data due to weak authentication, misconfigurations, or insider threats. Example: Leaked customer records due to a hacked cloud database.

2. Data Loss (1 Mark)

Accidental deletion, corruption, or loss of data due to system failures, human error, or cyberattacks. Example: Cloud storage provider crashes, losing critical backups.

3. Account Hijacking (1 Mark)

Attackers gain unauthorized access to cloud accounts through phishing, weak passwords, or credential leaks. Example: Hackers stealing API keys to manipulate cloud resources.

4. Insecure APIs (1 Mark)

Poorly secured cloud APIs can expose services to attacks such as SQL injection, cross-site scripting (XSS), and denial-of-service (DoS) attacks. Example: An exposed API key allowing attackers to access private data.

5. Denial-of-Service (DoS) Attacks (1 Mark)

Attackers overload cloud services, causing slowdowns or complete service outages. Example: A DDoS attack on a cloud-hosted website making it unavailable to users.

6. Insider Threats (1 Mark)

Employees or contractors with privileged access may misuse their permissions, leading to data theft or sabotage. Example: A disgruntled employee leaking confidential customer data.

7. Lack of Compliance (1 Mark)

Failure to meet data protection regulations (e.g., GDPR, HIPAA) can result in legal penalties and reputational damage. Example: A healthcare provider using an unsecured cloud storage system, violating HIPAA regulations.

To mitigate these threats, organizations must implement strong authentication, encryption, monitoring, and security best practices to ensure cloud security.

5 b. manually preparing or extending IT resources in response to workload fluctuation is time-intensive and unacceptably efficient. How can IT resources be scaled automatically in response to fluctuating demand? [8]

➢ Automatic Scaling of IT Resources in Response to Fluctuating Demand

Manual scaling of IT resources is slow and inefficient. To address this, **automatic scaling** (autoscaling) mechanisms ensure resources are allocated dynamically based on demand, improving efficiency and cost-effectiveness.

Methods of Automatic Scaling (4 Marks)

Vertical Scaling (Scaling Up/Down)

Increases or decreases the power (CPU, RAM) of an existing server.

Example: A cloud database increases RAM when queries spike and reduces it during low traffic.

Horizontal Scaling (Scaling Out/In)

Adds or removes instances (servers, containers) dynamically.

Example: An e-commerce website adds more web servers during sales events and removes them afterward.

Load Balancing

Distributes traffic across multiple servers to prevent overload.

Example: A cloud-based application directs requests to the least busy server.

Containerization & Orchestration (Kubernetes, Docker Swarm) Manages containers, ensuring applications scale automatically.

Example: Kubernetes automatically launches more containers when user requests increase.

Auto scaling Strategies (4 Marks)

Threshold-Based Scaling

Resources scale up or down based on pre-defined CPU, memory, or network usage limits.

Example: AWS Auto Scaling adds servers when CPU usage exceeds 80%.

Scheduled Scaling

Resources scale based on predictable demand patterns.

Example: An educational platform scales up at 9 AM when students log in and scales down at night.

Predictive Scaling (AI/ML-Based)

Uses machine learning to anticipate future demand and scale resources accordingly.

Example: Cloud AI predicts website traffic surges before a product launch and scales infrastructure proactively.

Event-Driven Scaling

Resources scale dynamically based on real-time triggers (e.g., user logins, order placements).

Example: A ride-sharing app scales up its backend when demand spikes during peak hours.

By implementing **auto scaling techniques**, organizations ensure **high performance, cost efficiency, and seamless user experience**, reducing manual intervention.

6a .Explain the key features and advantages of the Hadoop Distributed File System (HDFS) in the context of big data processing. [7].

- ➢ Key Features and Advantages of Hadoop Distributed File System (HDFS) in Big Data Processing

Hadoop Distributed File System (HDFS) is a **scalable, fault-tolerant, and highthroughput** storage system designed to handle **large-scale data processing** in **distributed computing environments**.

Key Features of HDFS (4 Marks)

Scalability – HDFS supports **horizontal scaling**, allowing data to be distributed across thousands of machines.

Fault Tolerance – Data is automatically **replicated (default: 3 copies)** across multiple nodes, ensuring recovery from failures.

High Throughput – Optimized for **batch processing**, enabling efficient handling of large files in **parallel**.

Write-Once, Read-Many Model – Once written, files cannot be modified, ensuring **data integrity and faster processing**.

Distributed Storage – Large files are **split into blocks (default: 128MB or 256MB)** and distributed across multiple nodes.

Master-Slave Architecture – The **Name Node** manages metadata, while **Data Nodes** store actual data.

Support for Large Files – HDFS is designed to handle **petabytes** of structured and unstructured data efficiently.

Advantages of HDFS in Big Data Processing (3 Marks)

Cost-Effective Storage – HDFS runs on commodity hardware, reducing infrastructure costs.

Efficient Data Locality – Processing occurs **close to data** (Data Locality Principle), reducing **network congestion**.

Seamless Integration with Big Data Tools – Works well with **MapReduce, Apache Spark, Hive, and Pig** for large-scale data analytics.

Conclusion:

HDFS is a **highly scalable, fault-tolerant, and efficient** file system that is essential for **big data processing**, making it ideal for **distributed computing environments**.

6b. Hadoop MapReduce Framework for Word Count

The **MapReduce framework** processes large datasets in a distributed manner. It consists of two main phases:

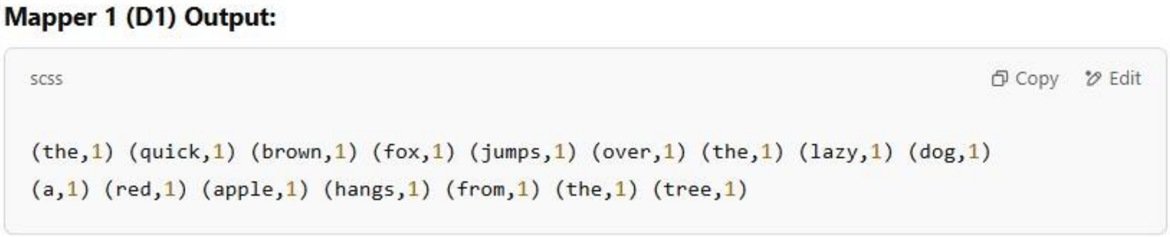
1. **Map Phase** – Splits input data and processes it in parallel.
2. **Reduce Phase** – Aggregates and summarizes the intermediate results.

Given Documents (Stored as Three HDFS Blocks)

- **D1:** "the quick brown fox jumps over the lazy dog. a red apple hangs from the tree."
- **D2:** "mountains cast long shadows during sunset. the tree provides shade for the red apple."
- **D3:** "the sunsets behind the mountains. the lazy dog barks at the quick brown fox."

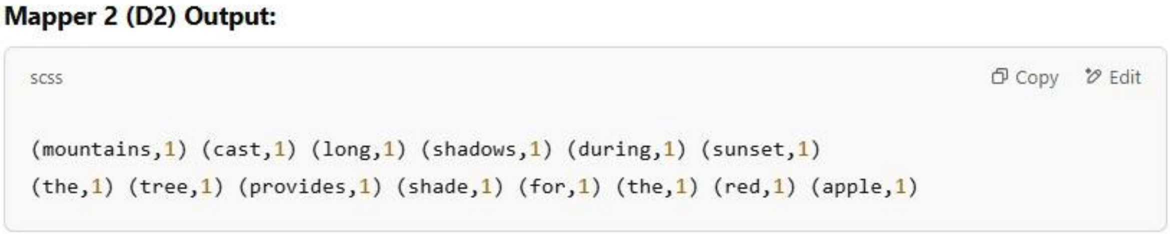
Step 1: Map Phase

Each **Mapper** processes a block and **emits (key, value) pairs**, where **key = word** and **value = 1**.



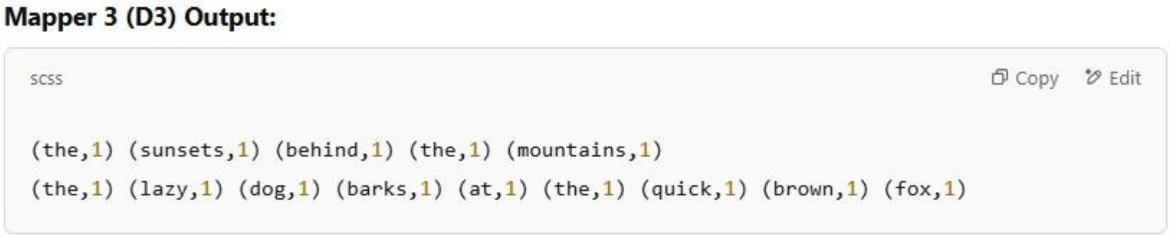
Mapper 1 (D1) Output:

```
(the,1) (quick,1) (brown,1) (fox,1) (jumps,1) (over,1) (the,1) (lazy,1) (dog,1)
(a,1) (red,1) (apple,1) (hangs,1) (from,1) (the,1) (tree,1)
```



Mapper 2 (D2) Output:

```
(mountains,1) (cast,1) (long,1) (shadows,1) (during,1) (sunset,1)
(the,1) (tree,1) (provides,1) (shade,1) (for,1) (the,1) (red,1) (apple,1)
```

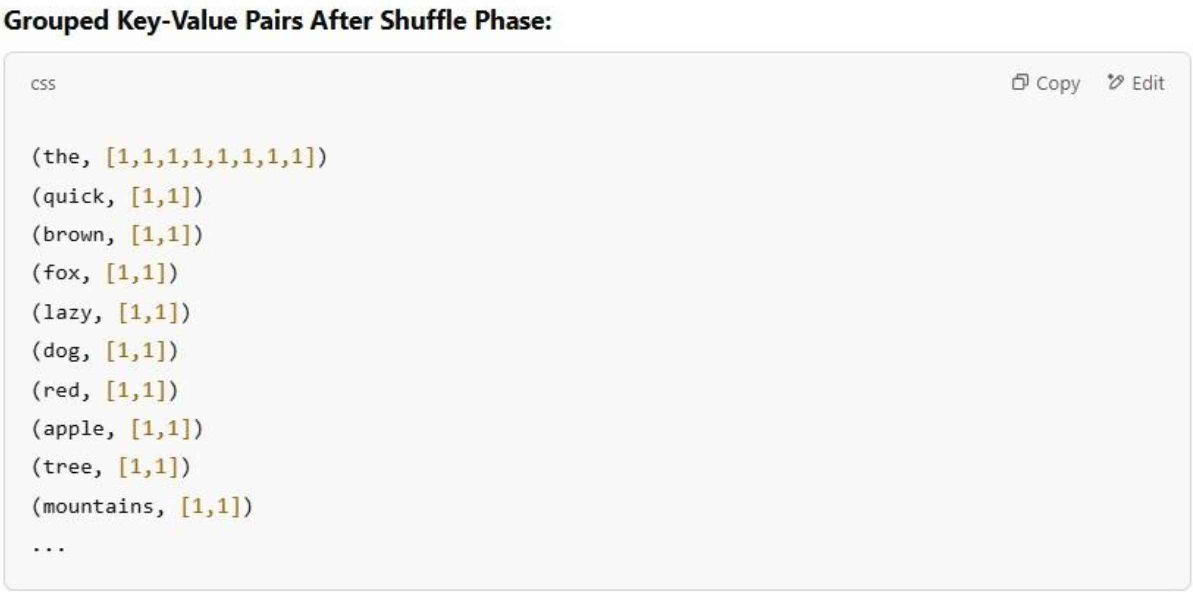


Mapper 3 (D3) Output:

```
(the,1) (sunsets,1) (behind,1) (the,1) (mountains,1)
(the,1) (lazy,1) (dog,1) (barks,1) (at,1) (the,1) (quick,1) (brown,1) (fox,1)
```

Step 2: Shuffle & Sort Phase

- The **Shuffle Phase** groups values by **key (word)** from all mappers.
- The **Sort Phase** sorts the words alphabetically.



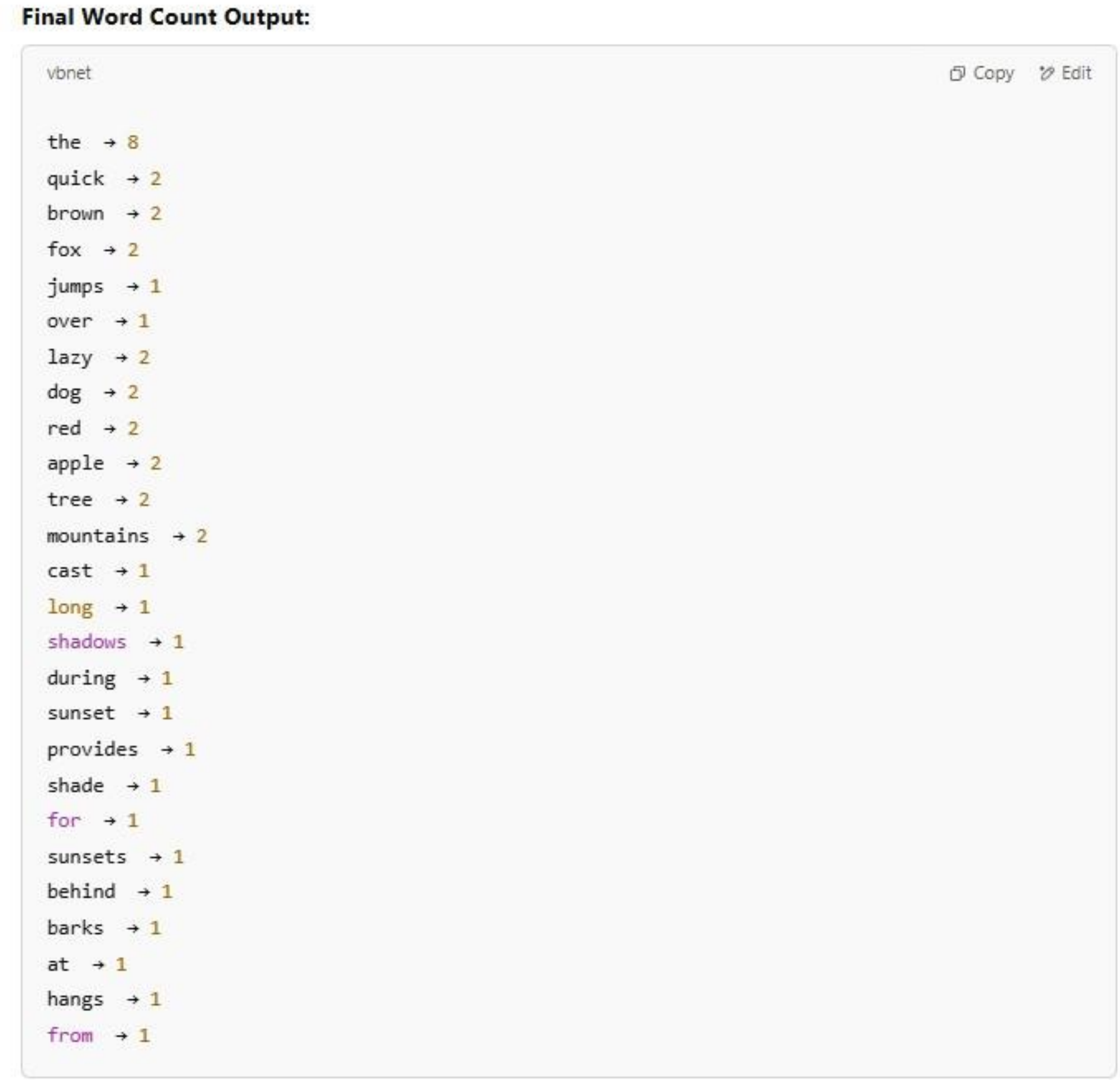
Grouped Key-Value Pairs After Shuffle Phase:

```
(the, [1,1,1,1,1,1,1,1])
(quick, [1,1])
(brown, [1,1])
(fox, [1,1])
(lazy, [1,1])
(dog, [1,1])
(red, [1,1])
(apple, [1,1])
(tree, [1,1])
(mountains, [1,1])
...
```

□

Step 3: Reduce Phase

Each **Reducer** sums up the word frequencies.



Final Word Count Output:

```
the  → 8
quick  → 2
brown  → 2
fox  → 2
jumps  → 1
over  → 1
lazy  → 2
dog  → 2
red  → 2
apple  → 2
tree  → 2
mountains  → 2
cast  → 1
long  → 1
shadows  → 1
during  → 1
sunset  → 1
provides  → 1
shade  → 1
for  → 1
sunsets  → 1
behind  → 1
barks  → 1
at  → 1
hangs  → 1
from  → 1
```

Conclusion

The Hadoop **MapReduce framework** successfully **counts word frequencies** by distributing tasks across multiple nodes using **Mappers, Shuffle & Sort, and Reducers**.

7. Write short note

1. Kubernetes (5 Marks)

Kubernetes is an open-source **container orchestration platform** that automates the deployment, scaling, and management of containerized applications. It works with tools like Docker and container runtimes to manage containers at scale.

Key Features of Kubernetes:

- **Pods:** A group of one or more containers that are deployed together on the same host machine.
- **Replication Controllers:** Ensures the desired number of pods are running.
- **Services:** Provides a stable IP address and DNS name for accessing pods.
- **Scaling:** Allows automatic scaling of applications based on demand.
- **Self-Healing:** Automatically restarts containers if they fail and reschedules them on healthy nodes.
- **Resource Management:** Efficiently manages resources like CPU and memory.

Advantages of Kubernetes:

- **Scalability:** Handles large-scale containerized workloads effectively.
- **Portability:** Supports multi-cloud and hybrid cloud deployments.
- **Automated Deployment & Updates:** Simplifies application deployment, scaling, and updates.
- **High Availability:** Ensures high availability of applications by distributing containers across nodes.

---

2. Identity and Access Management (IAM) (5 Marks)

Identity and Access Management (IAM) refers to the policies, processes, and technologies that ensure the right individuals (users, devices, applications) have the appropriate access to systems and data within an organization. IAM helps enforce security protocols and complies with regulations.

Key Components of IAM:

- **Authentication:** Verifies the identity of users, typically through passwords, biometrics, or multi-factor authentication (MFA).
- **Authorization:** Grants or denies access to resources based on the user's permissions or roles.
- **Roles & Permissions:** Defines the level of access a user has to different systems or data (e.g., admin, user, guest).

- **Single Sign-On (SSO):** Allows users to authenticate once and gain access to multiple systems without re-entering credentials.

- **Multi-Factor Authentication (MFA):** Requires users to provide additional authentication factors, enhancing security.

Advantages of IAM:

- **Improved Security:** Minimizes unauthorized access and prevents data breaches.
- **Compliance:** Helps meet regulatory requirements like GDPR and HIPAA.
- **Centralized Management:** Provides a single interface to manage access for multiple systems.
- **User Convenience:** Simplifies the login process for end-users, particularly with SSO and MFA.

---

3. Cloud Cube Model (5 Marks)

The **Cloud Cube Model** is a framework that helps businesses understand and evaluate cloud computing from **four key dimensions**:

1. **Internal vs. External:** Whether cloud resources are managed and deployed within the organization (private cloud) or externally (public cloud).
2. **Shared vs. Dedicated:** Whether the resources are shared among multiple tenants (multitenant) or dedicated to a single tenant.
3. **Online vs. Offline:** Whether the cloud services are always online (cloud-based) or temporarily offline (e.g., for backups).
4. **Integrating with Legacy Systems:** How easily the cloud services can integrate with existing on-premise infrastructure.

Advantages of Cloud Cube Model:

- **Helps Organizations Choose the Right Cloud Deployment:** Aids in selecting the appropriate level of control, security, and resource-sharing.
- **Flexible Cloud Options:** Provides a framework for understanding different cloud models (private, public, hybrid, and community clouds).
- **Simplifies Cloud Strategy:** Helps organizations assess their cloud needs in relation to cost, security, and scalability.

BOARD QUESTION 5 ota fark xa

1a. why could cloud computing be successful when other paradigms have failed? What are the advantages of cloud computing? 7

- Cloud computing has succeeded where other paradigms like grid computing and traditional IT infrastructures have failed due to the following reasons:

Scalability & Elasticity (1 Mark) – Unlike traditional IT infrastructure, cloud computing allows businesses to dynamically scale resources up or down based on demand.

Cost-Effectiveness (1 Mark) – The pay-as-you-go model eliminates the need for large upfront investments, reducing costs significantly.

Reliability & Disaster Recovery (1 Mark) – Cloud providers ensure high availability and redundancy, overcoming failures common in on-premise systems.

Security & Compliance (1 Mark) – Cloud providers implement strong security measures, making cloud computing safer than many traditional IT solutions.

---

2. Advantages of Cloud Computing *(3 Marks)*

On-Demand Self-Service (0.5 Marks) – Users can provision computing resources as needed without human intervention.

Broad Network Access (0.5 Marks) – Cloud services are accessible from various devices over the internet.

Resource Pooling (0.5 Marks) – Multiple users share resources efficiently, reducing operational costs.

Rapid Elasticity (0.5 Marks) – Cloud platforms quickly adjust to demand fluctuations.

Measured Service (0.5 Marks) – Users pay only for what they consume, improving cost efficiency.

Automatic Software Updates (0.5 Marks) – Providers handle system maintenance, reducing manual workload.

2a. what is Platform-as-a-Service (PaaS)? Suru ko question matra

Platform-as-a-Service (PaaS) is a cloud computing model that provides a platform for developers to build, test, deploy, and manage applications without worrying about the underlying infrastructure.

Example & Features (1 Mark):

It includes tools like databases, middleware, development frameworks, and runtime environments.

Examples: Google App Engine, Microsoft Azure App Services, and AWS Elastic Beanstalk.

7 a. Dockers

1. Introduction (1 Mark)
Docker is an open-source platform that allows developers to automate the deployment of applications inside lightweight, portable **containers**. These containers package applications with their dependencies, ensuring they run consistently across different environments.

2. Key Features (2 Marks)

Containerization (0.5 Mark): Packages applications and their dependencies into isolated containers.

Portability (0.5 Mark): Runs consistently across different environments (development, testing, production).

Lightweight (0.5 Mark): Uses fewer resources than virtual machines since containers share the host OS kernel.

Scalability (0.5 Mark): Easily scales applications using orchestration tools like Kubernetes.

3. Advantages (1 Mark)

Faster deployment and startup times.

Efficient resource utilization.

Simplifies DevOps workflows by enabling CI/CD (Continuous Integration/Continuous Deployment).

4. Example Use Case (1 Mark)
A company developing a web application can use Docker to package the app along with its required libraries, ensuring it runs the same way on any server or developer's machine.

7c. VPC

1. Definition (1 Mark)
A **Virtual Private Cloud (VPC)** is a private, isolated section of a public cloud where users can deploy resources securely. It allows organizations to define and control their networking environment, similar to an on-premise data center but with cloud benefits.

2. Key Features (2 Marks)

Isolation (0.5 Mark): Ensures secure separation of resources from other cloud users.

Customizable IP Addressing (0.5 Mark): Allows users to define private IP address ranges.

Subnets & Routing (0.5 Mark): Users can create subnets and configure routing for internal and external communication.

Security Controls (0.5 Mark): Provides security features like **firewalls, security groups, and access control lists (ACLs).**

3. Advantages (1 Mark)

Enhanced **security** with controlled access.

Improved **network performance** through private connectivity.

Scalability and **flexibility** for resource management.

4. Example Use Case (1 Mark)
A company hosting a **secure web application** can use a VPC to separate public-facing web servers from private databases, ensuring **data protection and controlled access**.