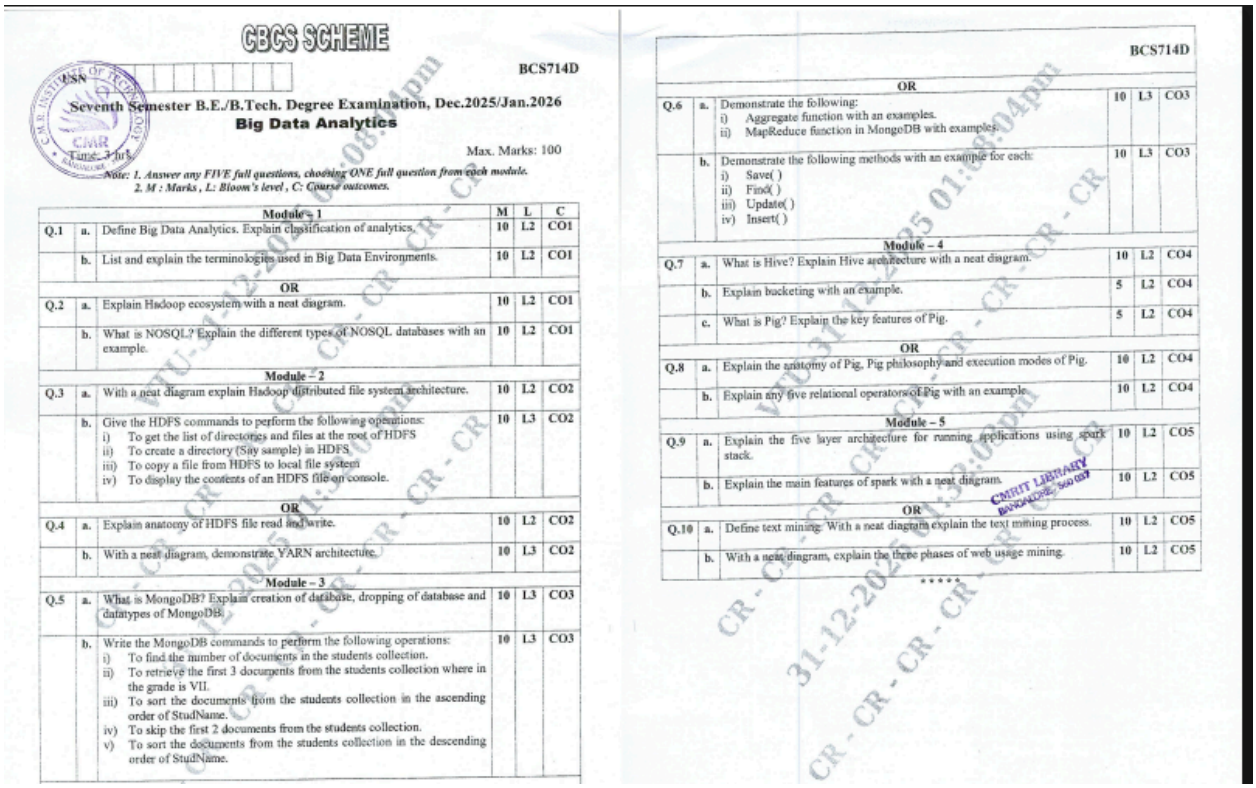


CBCS SCHEME

USN

BCS714D

Seventh Semester B.E./B.Tech. Degree Examination, Dec.2025/Jan.2026

Big Data Analytics

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.	Define Big Data Analytics. Explain classification of analytics.	10	L2	CO1
	b.	List and explain the terminologies used in Big Data Environments.	10	L2	CO1
		OR			
Q.2	a.	Explain Hadoop ecosystem with a neat diagram.	10	L2	CO1
	b.	What is NOSQL? Explain the different types of NOSQL databases with an example.	10	L2	CO1
		Module – 2			
Q.3	a.	With a neat diagram explain Hadoop distributed file system architecture.	10	L2	CO2
	b.	Give the HDFS commands to perform the following operations: i) To get the list of directories and files at the root of HDFS ii) To create a directory (Say sample) in HDFS iii) To copy a file from HDFS to local file system iv) To display the contents of an HDFS file on console.	10	L3	CO2
		OR			
Q.4	a.	Explain anatomy of HDFS file read and write.	10	L2	CO2
	b.	With a neat diagram, demonstrate YARN architecture.	10	L3	CO2
		Module – 3			
Q.5	a.	What is MongoDB? Explain creation of database, dropping of database and datatypes of MongoDB.	10	L3	CO3
	b.	Write the MongoDB commands to perform the following operations: i) To find the number of documents in the students collection. ii) To retrieve the first 3 documents from the students collection where in the grade is VII. iii) To sort the documents from the students collection in the ascending order of StudName. iv) To skip the first 2 documents from the students collection. v) To sort the documents from the students collection in the descending order of StudName.	10	L3	CO3

BCS714D

		OR			
Q.6	a.	Demonstrate the following: i) Aggregate function with an examples. ii) MapReduce function in MongoDB with examples.	10	L3	CO3
	b.	Demonstrate the following methods with an example for each: i) Save( ) ii) Find( ) iii) Update( ) iv) Insert( )	10	L3	CO3
		Module – 4			
Q.7	a.	What is Hive? Explain Hive architecture with a neat diagram.	10	L2	CO4
	b.	Explain bucketing with an example.	5	L2	CO4
	c.	What is Pig? Explain the key features of Pig.	5	L2	CO4
		OR			
Q.8	a.	Explain the anatomy of Pig, Pig philosophy and execution modes of Pig.	10	L2	CO4
	b.	Explain any five relational operators of Pig with an example.	10	L2	CO4
		Module – 5			
Q.9	a.	Explain the five layer architecture for running applications using spark stack.	10	L2	CO5
	b.	Explain the main features of spark with a neat diagram.	10	L2	CO5
		OR			
Q.10	a.	Define text mining. With a neat diagram explain the text mining process.	10	L2	CO5
	b.	With a neat diagram, explain the three phases of web usage mining.	10	L2	CO5

*****

DEC2025-JAN2026 EXAMINATION: VTU
SOLUTION: BIG DATA ANALYTICS

			Module – 1	M	L	C
Q.1	a.	Define Big Data Analytics. Explain classification of analytics.		10	L2	CO1
	b.	List and explain the terminologies used in Big Data Environments.		10	L2	CO1

Q1

ANSWER

Definition of Big Data Analytics

Big Data Analytics refers to the process of examining **large, complex, and diverse datasets** (structured, semi-structured, and unstructured data) to uncover hidden patterns, correlations, trends, and useful information for decision-making.

It uses advanced tools and techniques such as:

- Data Mining
- Machine Learning
- Statistical Analysis
- Predictive Modeling

Characteristics of Big Data (5 V's)

1. **Volume** – Huge amount of data
2. **Velocity** – Speed of data generation
3. **Variety** – Different types of data
4. **Veracity** – Data uncertainty/quality
5. **Value** – Extracting useful insights

Classification of Analytics

Big Data Analytics is broadly classified into four types:

1. Descriptive Analytics – "What happened?"

- Analyzes past data
- Summarizes historical information
- Uses reports, dashboards, visualization

Example:
Monthly sales report, website traffic summary

2. Diagnostic Analytics – "Why did it happen?"

- Finds root cause of problems
- Uses drill-down, correlation, data mining

Example:
Why sales dropped in a particular region

3. Predictive Analytics – "What will happen?"

- Uses historical data to forecast future outcomes
- Uses machine learning, regression, time-series models

Example:
Predicting customer churn, stock market trends

4. Prescriptive Analytics – "What should we do?"

- Suggests best possible actions
- Uses optimization and simulation techniques

Example:
Recommending best pricing strategy

Diagram Representation (Flow)

Descriptive → Diagnostic → Predictive → Prescriptive
(Past → Cause → Future → Decision)

Q1 (b) List and Explain Terminologies Used in Big Data Environments

Below are important terminologies in Big Data:

1. Structured Data

- Organized in rows and columns
- Stored in RDBMS

- Example: Banking records

2. Unstructured Data

- No predefined format
- Example: Images, videos, social media posts

3. Semi-Structured Data

- Has partial structure
- Example: XML, JSON files

4. Data Warehouse

- Centralized repository of structured data
- Used for reporting and analysis

5. Data Lake

- Stores raw data in native format
- Supports structured and unstructured data

6. Hadoop

- Open-source framework for distributed storage and processing
- Handles large datasets

7. HDFS (Hadoop Distributed File System)

- Distributed storage system in Hadoop
- Stores data across multiple machines

8. MapReduce

- Programming model for processing large datasets in parallel

9. NoSQL

- Non-relational databases
- Handles unstructured data
- Example: MongoDB, Cassandra

10. Spark

- Fast data processing engine
- Supports real-time analytics

11. ETL (Extract, Transform, Load)

- Process of extracting data, transforming it, and loading into system

12. Data Mining

- Process of discovering patterns from large datasets

13. Machine Learning

- Algorithms that learn from data to make predictions

14. Real-Time Processing

- Processing data immediately as it is generated

15. Batch Processing

- Processing data in large chunks at scheduled intervals

Big Data Analytics enables organizations to process large volumes of complex data efficiently. The classification of analytics helps businesses move from understanding past events to making intelligent future decisions. Big Data terminologies provide foundational knowledge required to work in modern data environments.

Q.2	a.	Explain Hadoop ecosystem with a neat diagram.	10	L2	CO1
	b.	What is NOSQL? Explain the different types of NOSQL databases with an example.	10	L2	CO1

Q2:
Answer-

Q2 (a) Explain Hadoop Ecosystem with a Neat Diagram

Introduction to Hadoop

Apache Hadoop is an open-source framework used for **distributed storage and processing of large datasets** across clusters of computers using simple programming models.

It is designed to scale up from single servers to thousands of machines.

Core Components of Hadoop

1. HDFS (Hadoop Distributed File System)

- Distributed storage system
- Stores large files across multiple machines
- Fault tolerant

Components:

- **NameNode** – Master node (manages metadata)
- **DataNode** – Slave nodes (store actual data)

2. MapReduce

- Programming model for parallel data processing
- Works in two phases:
    - **Map phase** – Processes input data
    - **Reduce phase** – Aggregates results

3. YARN (Yet Another Resource Negotiator)

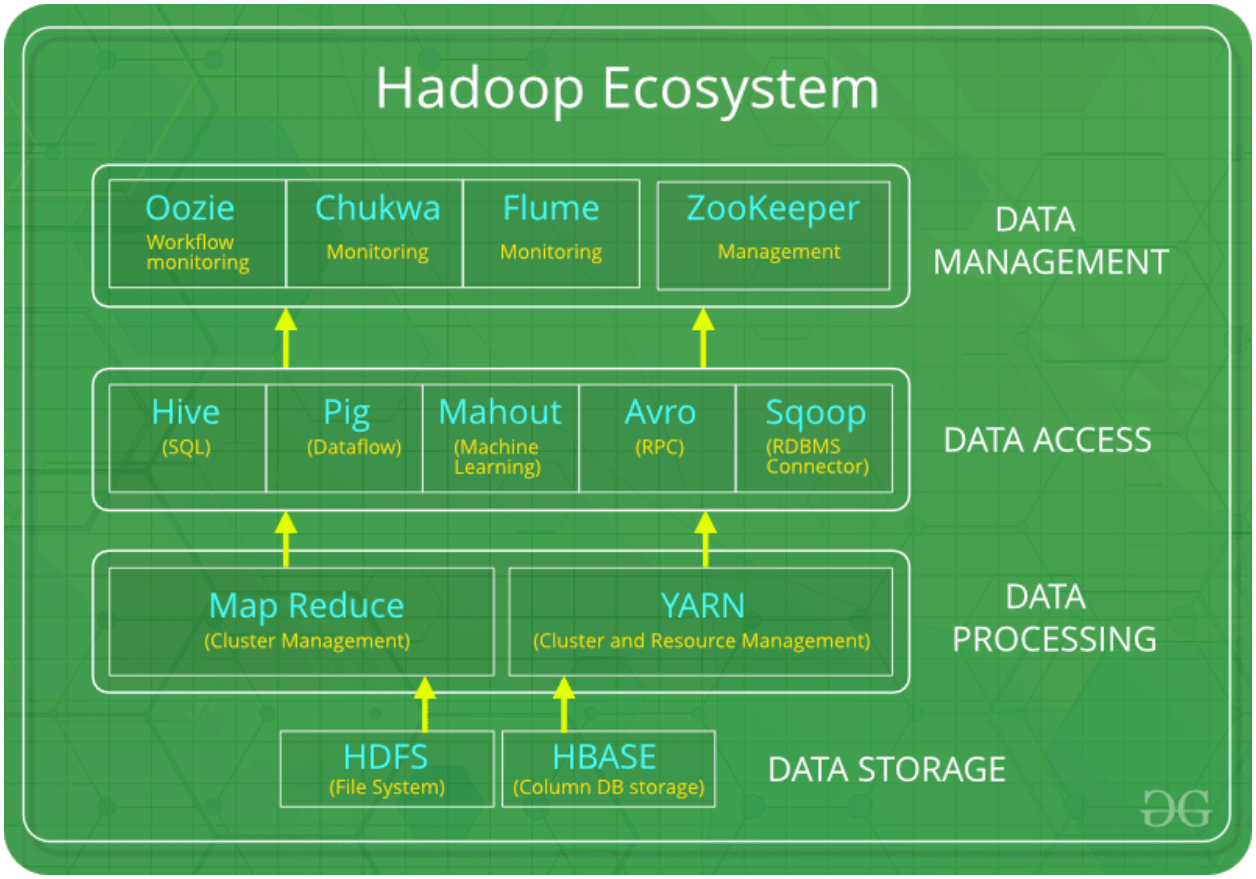
- Resource management layer
- Allocates system resources to applications
- Improves scalability

Hadoop Ecosystem Components

Hadoop ecosystem includes several tools that support storage, processing, querying, and data management.

Component	Purpose
Hive	Data warehousing & SQL-like queries
Pig	High-level scripting for data analysis
HBase	NoSQL database on HDFS
Sqoop	Transfer data between RDBMS & Hadoop

Flume	Collects log and streaming data
Spark	Fast in-memory data processing
Oozie	Workflow scheduler
Zookeeper	Coordination service



Advantages of Hadoop

- Scalability
- Fault tolerance

- Cost-effective
- Handles structured & unstructured data

Conclusion

Hadoop ecosystem provides a complete platform for storing, processing, and analyzing big data in a distributed environment.

Q2 (b) What is NoSQL? Explain the Different Types of NoSQL Databases with Example

Definition of NoSQL

NoSQL stands for "Not Only SQL".
It is a **non-relational database system** designed to handle large volumes of unstructured and semi-structured data.

Unlike traditional RDBMS:

- Does not use fixed schema
- Scales horizontally
- Supports distributed architecture

Characteristics of NoSQL

- Schema-less
- High scalability
- High performance
- Supports big data applications

Types of NoSQL Databases

NoSQL databases are mainly classified into **four types**:

1. Key-Value Stores

- Data stored as key-value pairs
- Simple and fast

Example:
Redis

Use Case:
Caching systems, session management

2. Document-Oriented Databases

- Data stored in JSON-like documents
- Flexible schema

Example:
MongoDB

Use Case:
Content management systems, e-commerce

3. Column-Family Stores

- Data stored in columns instead of rows
- Suitable for large distributed systems

Example:
Apache Cassandra

Use Case:
Real-time analytics, IoT applications

4. Graph Databases

- Data stored as nodes and edges
- Used for relationship-based queries

Example:
Neo4j

Use Case:
Social networks, recommendation systems

Comparison Table

Type	Structure	Example	Best For
Key-Value	Key + Value	Redis	Caching
Document	JSON Documents	MongoDB	Web apps
Column	Column families	Cassandra	Big data
Graph	Nodes & Edges	Neo4j	Relationship data

NoSQL databases are designed to manage large-scale, distributed, and unstructured data efficiently. They provide flexibility, scalability, and high performance compared to traditional relational databases.

Q3:

Q.3	a.	With a neat diagram explain Hadoop distributed file system architecture.	10	L2	CO2
	b.	Give the HDFS commands to perform the following operations: i) To get the list of directories and files at the root of HDFS ii) To create a directory (Say sample) in HDFS iii) To copy a file from HDFS to local file system iv) To display the contents of an HDFS file on console.	10	L3	CO2

Answer-

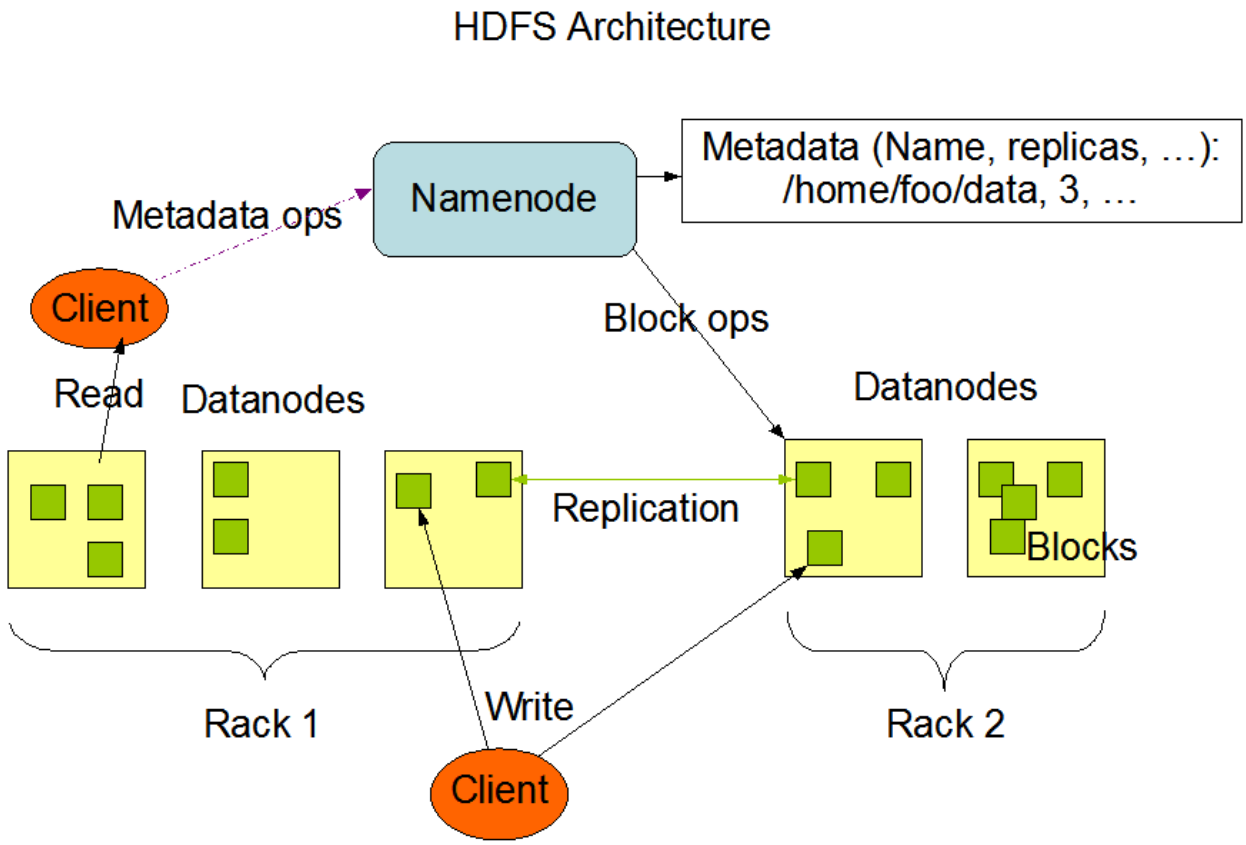
Q3 (a) With a neat diagram explain Hadoop Distributed File System (HDFS) Architecture

Introduction

Hadoop Distributed File System (HDFS) is the primary storage system of Apache Hadoop.
It is designed to store very large files across multiple machines in a distributed environment.

HDFS follows **Master–Slave architecture**.

HDFS Architecture Components



1. NameNode (Master Node)

- Manages metadata (file names, permissions, block locations)

- Controls access to files
- Maintains file system namespace
- Only one active NameNode (in basic architecture)

2. DataNode (Slave Nodes)

- Stores actual data blocks
- Performs read/write operations
- Sends heartbeat signals to NameNode
- Multiple DataNodes exist in cluster

3. Secondary NameNode

- Not a backup NameNode
- Periodically merges metadata (FsImage + EditLog)

Working of HDFS

1. File is divided into **blocks** (default 128 MB)
2. Blocks are stored on different DataNodes
3. Each block is replicated (default replication factor = 3)
4. Ensures **fault tolerance**

Features of HDFS

- Distributed storage

- Fault tolerance
- High throughput
- Scalable
- Cost-effective

Advantages

✔ Suitable for large files
✔ Automatic replication
✔ Reliable data storage

HDFS provides reliable, scalable, and distributed storage for big data applications using master-slave architecture.

Q3 (b) Give the HDFS commands for the following operations

All commands use:

hdfs dfs <command>

i) To get the list of directories and files at the root of HDFS

hdfs dfs -ls /

ii) To create a directory (say sample) in HDFS

hdfs dfs -mkdir /sample

(If parent directories are needed:)

hdfs dfs -mkdir -p /sample

iii) To copy a file from HDFS to local file system

hdfs dfs -get /sample/file.txt /home/user/

OR

hdfs dfs -copyToLocal /sample/file.txt /home/user/

iv) To display the contents of an HDFS file on console

hdfs dfs -cat /sample/file.txt

OR (for large files)

hdfs dfs -tail /sample/file.txt

Summary Table

Operation	Command
List root directory	hdfs dfs -ls /
Create directory	hdfs dfs -mkdir /sample
Copy to local	hdfs dfs -get source destination
Display file	hdfs dfs -cat filename

Q4:

Q.4	a.	Explain anatomy of HDFS file read and write.	10	L2	CO2
	b.	With a neat diagram, demonstrate YARN architecture.	10	L3	CO2

ANSWER:

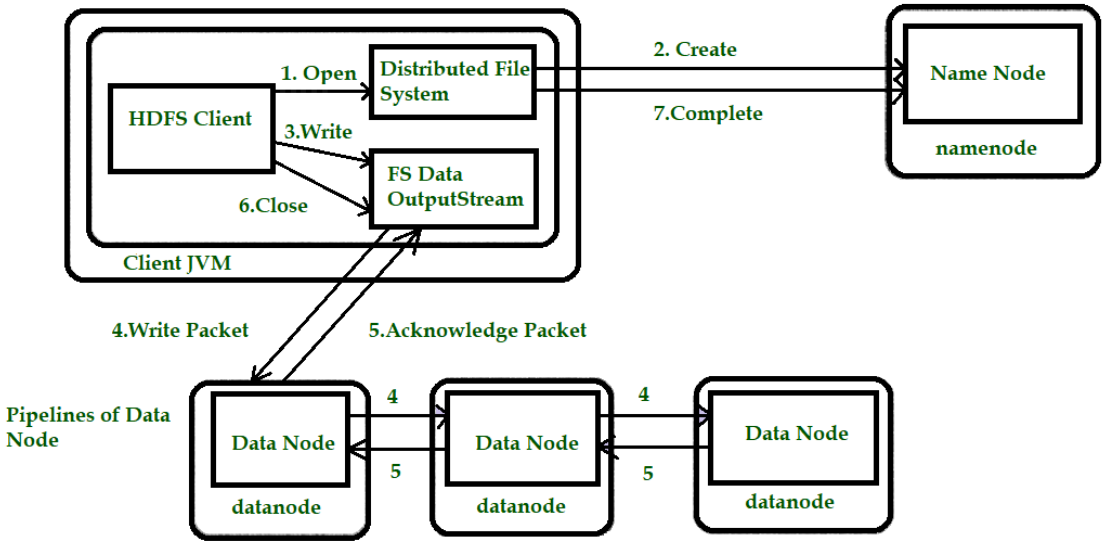
Q4 (a) Explain Anatomy of HDFS File Read and Write

Hadoop Distributed File System (HDFS) manages file read and write operations in a distributed and fault-tolerant manner.

It involves:

- Client
- NameNode
- DataNodes

HDFS Write Operation (File Upload Process)



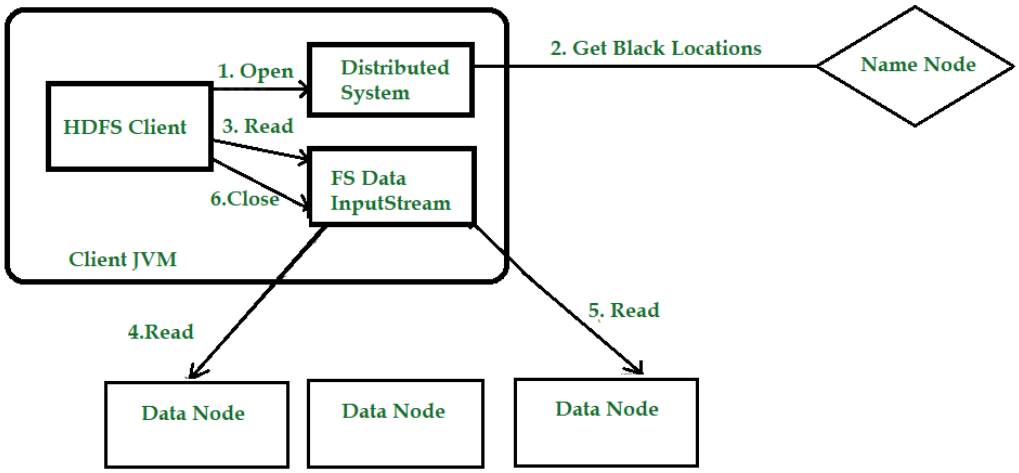
Step-by-Step Process

1. **Client Request**
    - Client sends file creation request to NameNode.
2. **Metadata Check**

- NameNode checks permissions and whether file already exists.

3. **Block Allocation**
    - NameNode divides file into blocks (default 128 MB).
    - Selects DataNodes for replication (default replication factor = 3).
4. **Data Transfer**
    - Client writes data to first DataNode.
    - First DataNode forwards to second.
    - Second forwards to third (Pipeline mechanism).
5. **Acknowledgment**
    - Acknowledgment flows back in reverse order.
    - If successful, next block is written.
6. **File Close**
    - NameNode updates metadata after completion.
    -

HDFS Read Operation (File Download Process)



HDFS Read Operation (File Download Process)

Step-by-Step Process

1. Client sends read request to NameNode.
2. NameNode provides block locations.
3. Client connects directly to nearest DataNode.
4. Data is read block-by-block.
5. If DataNode fails, client switches to another replica.

Key Features

✔ Pipeline replication
✔ Fault tolerance
✔ High throughput
✔ Rack awareness

Conclusion

HDFS ensures reliable file storage by dividing files into blocks, replicating them, and using efficient read/write mechanisms.

Q4 (b) With a neat diagram, demonstrate YARN Architecture

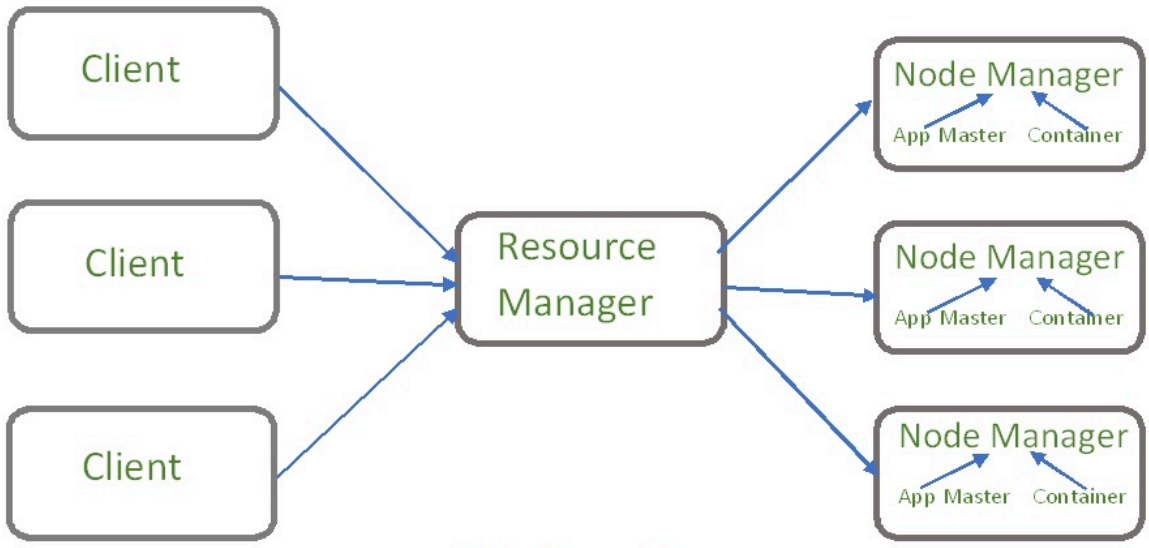
Introduction

Apache Hadoop uses
Apache Hadoop YARN (Yet Another Resource Negotiator) for resource management and job scheduling.

YARN separates:

- Resource Management
- Job Processing

Main Components of YARN



Hadoop Yarn architecture

1 ResourceManager (RM)

- Global resource manager
- Allocates resources to applications

Contains:

- Scheduler
- Applications Manager

2 NodeManager (NM)

- Runs on each worker node

- Manages containers
- Reports to ResourceManager

3️⃣ ApplicationMaster (AM)

- Created per application
- Manages application lifecycle
- Negotiates resources

4️⃣ Container

- Bundle of resources (CPU, RAM)
- Runs application tasks

Working of YARN

1. Client submits job to ResourceManager.
2. RM allocates container for ApplicationMaster.
3. AM negotiates resources.
4. NodeManagers launch containers.
5. Tasks execute and report status.

Advantages of YARN

✔ Better cluster utilization
✔ Scalability
✔ Multi-application support
✔ Supports MapReduce, Spark, etc.

Conclusion

YARN improves Hadoop by separating resource management from processing, making the system scalable and efficient.

Q5:

Q.5	a.	What is MongoDB? Explain creation of database, dropping of database and datatypes of MongoDB.	10	L3	CO3
	b.	Write the MongoDB commands to perform the following operations: i) To find the number of documents in the students collection. ii) To retrieve the first 3 documents from the students collection where in the grade is VII. iii) To sort the documents from the students collection in the ascending order of StudName. iv) To skip the first 2 documents from the students collection. v) To sort the documents from the students collection in the descending order of StudName.	10	L3	CO3

ANSWER:

Q5 (a) What is MongoDB? Explain creation of database, dropping of database and datatypes of MongoDB

Introduction

MongoDB is a **NoSQL document-oriented database** that stores data in flexible, JSON-like documents (BSON format).

It is:

- Schema-less
- Scalable
- High performance
- Suitable for Big Data applications

Features of MongoDB

✔ Document-based storage
✔ Dynamic schema
✔ Horizontal scaling
✔ High availability
✔ Indexing support

1️⃣ Creation of Database in MongoDB

MongoDB creates a database automatically when data is inserted.

Command to Create / Switch Database

use studentDB

If database does not exist, it will be created when you insert data.

Insert Data (to create DB physically)

db.students.insertOne({name:"Anita", grade:"VI"})

2️⃣ Dropping (Deleting) a Database

Command

db.dropDatabase()

This deletes the current database completely.

3️⃣ MongoDB Data Types

MongoDB supports BSON (Binary JSON) data types:

Data Type	Description
String	Text values
Integer	Numeric values

Double	Decimal numbers
Boolean	true / false
Date	Date values
Array	List of values
Object	Embedded document
ObjectId	Unique ID
Null	Null value
Timestamp	Time-based value

Example Document:

```
{
_id: ObjectId("1234"),
name: "Ravi",
age: 12,
grade: "VI",
subjects: ["Math", "Science"],
```

active: true

}

MongoDB is a flexible NoSQL database that allows easy creation, deletion, and management of large volumes of unstructured data.

Q5 (b) Write MongoDB commands for the following operations

Assume collection name: **students**

i) To find the number of documents in the students collection

db.students.countDocuments()

OR (older method)

db.students.count()

ii) To retrieve the first 3 documents from the students collection where grade is VI

db.students.find({grade:"VI"}).limit(3)

iii) To sort the documents from the students collection in ascending order of StudName

db.students.find().sort({StudName:1})

(1 = Ascending)

iv) To skip the first 2 documents from the students collection

db.students.find().skip(2)

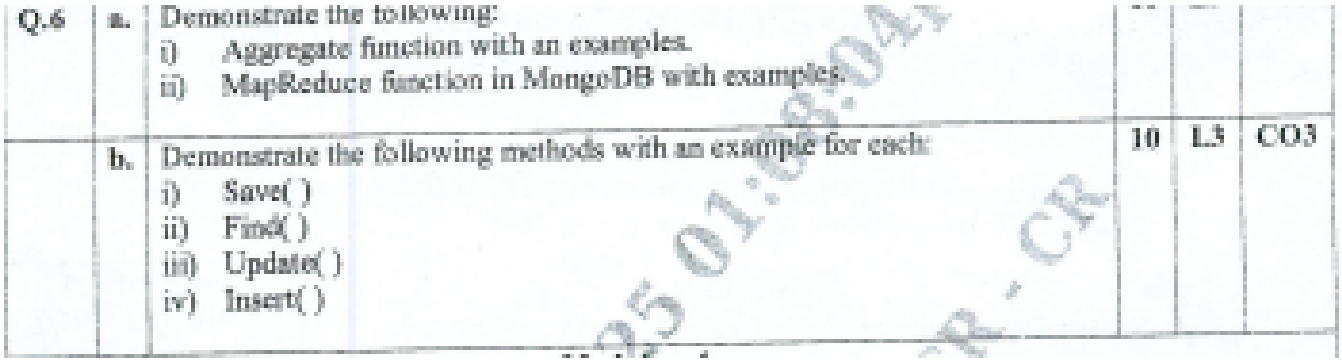
v) To sort the documents from the students collection in descending order of StudName

db.students.find().sort({StudName:-1})

(-1 = Descending)

Summary Table

Operation	MongoDB Command
Count documents	db.students.countDocuments()
First 3 grade VI	find({grade:"VI"}).limit(3)
Sort ascending	sort({StudName:1})
Skip 2 docs	skip(2)
Sort descending	sort({StudName:-1})



Q.6	a.	Demonstrate the following: i) Aggregate function with an examples. ii) MapReduce function in MongoDB with examples.			
	b.	Demonstrate the following methods with an example for each: i) Save( ) ii) Find( ) iii) Update( ) iv) Insert( )	10	L3	CO3

a) Demonstrate the following methods with an example for each

---

i) Aggregate Function in MongoDB

Aggregation is used to process data records and return computed results (like sum, average, count, grouping, etc.).

- **Example: Find total salary department-wise**

Collection: employees

```
{ "name": "A", "dept": "IT", "salary": 50000 }
{ "name": "B", "dept": "IT", "salary": 60000 }
{ "name": "C", "dept": "HR", "salary": 40000 }
```

Query:

```
db.employees.aggregate([
  { $group: { _id: "$dept", totalSalary: { $sum: "$salary" } } }
])
```

Output:

```
{ "_id": "IT", "totalSalary": 110000 }
{ "_id": "HR", "totalSalary": 40000 }
```

$group groups documents by department
$sum calculates total salary

---

ii) MapReduce Function in MongoDB

MapReduce is used for large-scale data processing.

It consists of:

- **Map function** → emits key-value pairs
- **Reduce function** → aggregates values

Example: Count number of employees in each department

Map Function:

```
var mapFunction = function() {
  emit(this.dept, 1);
};
```

Reduce Function:

```
var reduceFunction = function(key, values) {
  return Array.sum(values);
};
```

Execute:

```
db.employees.mapReduce(
  mapFunction,
  reduceFunction,
  { out: "dept_count" }
)
```

Output stored in dept_count collection
Gives count of employees per department

---

b) Demonstrate the following methods with examples

---

i) save()

Used to insert or update a document.

Example:

```
db.students.save({ _id: 1, name: "Ravi", marks: 85 })
```

If _id:1 exists → it updates
If not → it inserts

---

ii) find()

Used to retrieve documents.

Example:

```
db.students.find()
```

Displays all records.

With condition:

```
db.students.find({ marks: { $gt: 80 } })
```

Displays students with marks > 80.

---

iii) update()

Used to update existing documents.

Example:

```
db.students.update(
  { name: "Ravi" },
  { $set: { marks: 90 } }
)
```

Updates Ravi’s marks to 90.

---

iv) insert()

Used to insert new documents.

Example:

```
db.students.insert({
```

```
  name: "Anita",
  marks: 88
})
```

Adds new record to collection.

Q.7	a.	What is Hive? Explain Hive architecture with a neat diagram.	10	L2	CO4
	b.	Explain bucketing with an example.	5	L2	CO4
	c.	What is Pig? Explain the key features of Pig.	5	L2	CO4
		OR			

Hive was created by Facebook. Hive is a data warehousing tool and is also a data store on the top of Hadoop. An enterprise uses a data warehouse as large data repositories that are designed to enable the tracking, managing, and analyzing the data.

Components of Hive architecture are:

Hive Server (Thrift) - An optional service that allows a remote client to submit requests to

Hive and retrieve results. Requests can use a variety of programming languages.

Thrift Server exposes a very simple client API toexecute HiveQL statements.

Hive CLI (Command Line Interface) - Popular interface to interact with Hive.

Hive runs in local mode that uses local storage when running the CLI on a Hadoop cluster instead of HDFS.

Web Interface - Hive can be accessed using a web browser as well. This requires a HWI Server running on some designated code. The URL http:// hadoop:<port no.> / hwi command can be used to access Hive through the web.

Metastore - It is the system catalog. All other components of Hive interact with the Metastore. It stores the schema or metadata of tables, databases, columns in a table, their data types and HDFS mapping.

Hive Driver - It manages the life cycle of a HiveQL statement during compilation, optimization and execution.

HIVE ARCHITECTURE

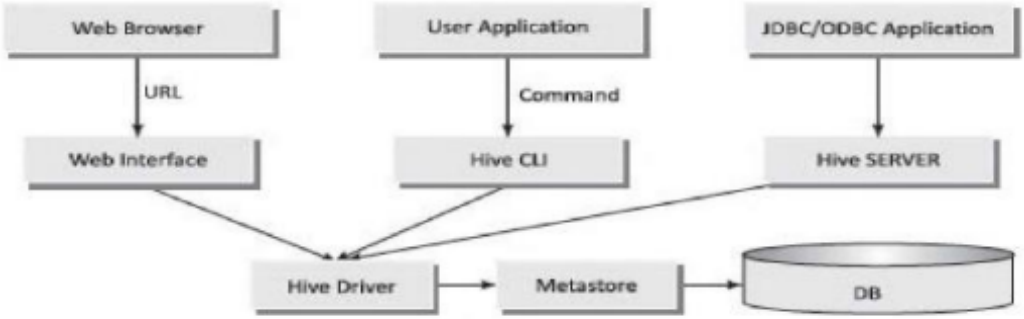


Figure 4.10 Hive architecture

(B)

Bucketing is a technique in MongoDB where related data values are grouped into a single document instead of storing each record separately.

It is mainly used in:

- Time-series data
- IoT sensor data
- Log data
- Financial transactions

Instead of storing one document per event, we store multiple related values inside an array in one document (a “bucket”).

Example With Bucketing

Instead of separate documents:

```
{
 "sensorId": 101,
 "date": "2026-02-25",
 "readings": [
    { "time": "10:00", "temp": 30 },
```

```
    { "time": "10:01", "temp": 31 },
    { "time": "10:02", "temp": 29 }
  ]
}
```

Now:

- One document stores many readings
- Fewer documents in collection

©
PIG

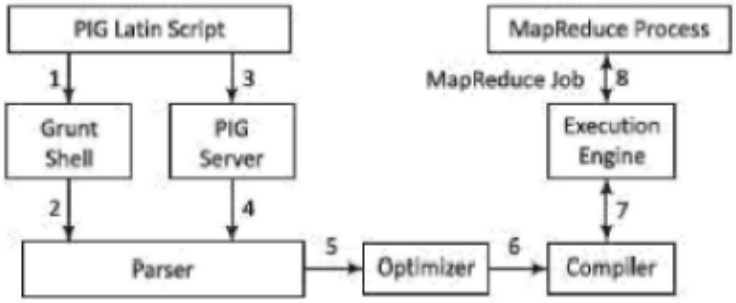
Apache Pig is a high-level data processing platform used for analyzing large datasets in **Hadoop**.

It provides a scripting language called **Pig Latin** to write data processing programs easily.

- It is an abstract over MapReduce
- It is an execution framework for parallel processing
- Reduces the complexities of writing a MapReduce program
- Is a high-level dataflow language. Dataflow language means that a Pig
- operation node takes the inputs and generates the output for the next node
- Is mostly used in HDFS environment
- Performs data manipulation operations at files at data nodes in Hadoop.
- Applications of Apache Pig
- Analyzing large datasets
- Executing tasks involving adhoc processing
- Processing large data sources such as web logs and streaming online data
- Data processing for search platforms. Pig processes different types of data
- Processing time sensitive data loads; data extracts and analyzes quickly.

Q.8	a.	Explain the anatomy of Pig, Pig philosophy and execution modes of Pig.	10	L2	CO4
	b.	Explain any five relational operators of Pig with an example.	10	L2	CO4

Pig Architecture



The three ways to execute scripts are:

1. Grunt Shell: An interactive shell of Pig that executes the scripts.

2. Script File: Pig commands written in a script file that execute at Pig Server.

3. Embedded Script: Create UDFs for the functions unavailable in Pig built- in

operators. UDF can be used in other programming languages. The UDFs can embed

in Pig Latin Script file.

The three ways to execute scripts are:

1. Grunt Shell: An interactive shell of Pig that executes the scripts.

2. Script File: Pig commands written in a script file that execute at Pig Server.

3. Embedded Script: Create UDFs for the functions unavailable in Pig built- in

operators. UDF can be used in other programming languages. The UDFs can embed

in Pig Latin Script file.

The three ways to execute scripts are:

1. Grunt Shell: An interactive shell of Pig that executes the scripts.

2. Script File: Pig commands written in a script file that execute at Pig Server.

3. Embedded Script: Create UDFs for the functions unavailable in Pig built- in

operators. UDF can be used in other programming languages. The UDFs can embed in Pig Latin Script file.

Pig Philosophy

Pig was designed with the following philosophy:

✔ **Ease of Programming**

Writing MapReduce in Java is complex. Pig simplifies it.

✔ **Procedural Language**

Pig Latin allows step-by-step data transformation.

✔ **Handles Semi-Structured Data**

Works well with logs, JSON, etc.

✔ **Extensible**

Supports User Defined Functions (UDFs).

✔ **Optimization is Automatic**

User focuses on logic; Pig optimizes execution.

Execution Modes of Pig

Pig runs in two modes:

1. Local Mode

- Runs on single machine
- Uses local file system
- Used for testing and small datasets

Command:

```
pig -x local
```

2. MapReduce Mode (Hadoop Mode)

- Runs on Hadoop cluster
- Uses HDFS
- Used for large datasets

Command:

```
pig -x mapreduce
```

8 (b) Explain any five relational operators of Pig with an example

Relational operators in Pig are used to process relations (datasets).

LOAD

Loads data from file.

```
A = LOAD 'data.txt' USING PigStorage(',')
    AS (id:int, name:chararray, marks:int);
```

FILTER

Selects records based on condition.

```
B = FILTER A BY marks > 80;
```

GROUP

Groups data by a field.

```
C = GROUP A BY marks;
```

JOIN

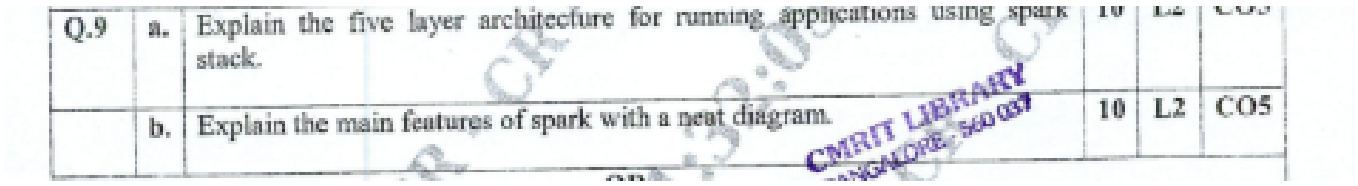
Joins two relations.

D = JOIN A BY id, B BY id;

FOREACH

Used to generate new relations from existing ones.

E = FOREACH A GENERATE name, marks;



Q.9	a.	Explain the five layer architecture for running applications using spark stack.	10	L2	CO5
	b.	Explain the main features of spark with a neat diagram.	10	L2	CO5

Application Layer

- User applications written in:
    - Scala
    - Python
    - Java
    - R
- These applications use Spark APIs.

Example:

- Machine learning
- Data analytics
- Stream processing

Spark Core Layer

This is the foundation of Spark.

Features:

- Task scheduling
- Memory management
- Fault recovery
- RDD (Resilient Distributed Dataset) support

Spark Core handles distributed execution.

Spark Libraries Layer

Provides built-in libraries:

- Spark SQL → Structured data processing
- Spark Streaming → Real-time data
- MLlib → Machine learning
- GraphX → Graph processing

These libraries run on top of Spark Core.

Cluster Manager Layer

Responsible for resource allocation.

Spark can run on:

- Standalone Cluster
- **Apache Hadoop YARN**
- **Apache Mesos**
- Kubernetes

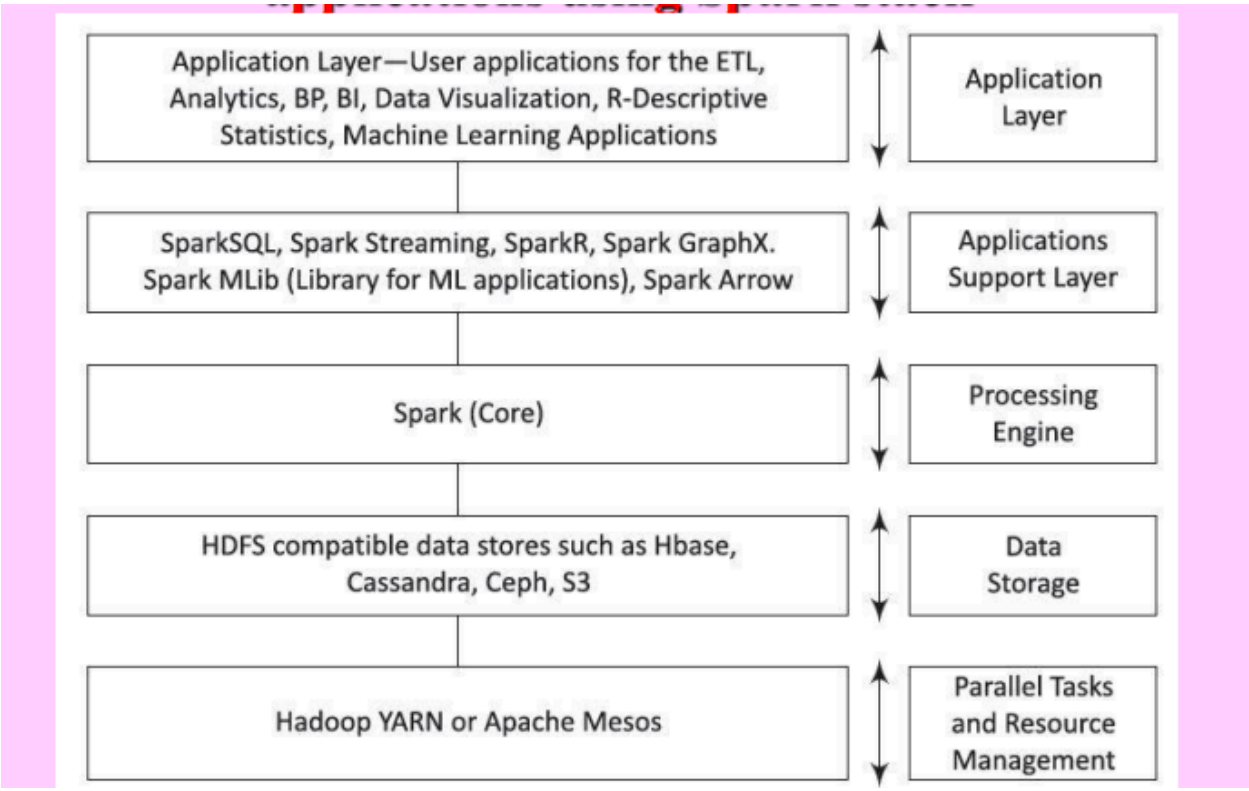
It manages CPU and memory across nodes.

Storage Layer

Spark reads/writes data from:

- HDFS
- Local file system
- Amazon S3
- HBase

Often integrated with **Apache Hadoop** storage.



9 (b) Explain the Main Features of Spark with a Neat Diagram

In-Memory Processing

- Stores intermediate data in memory.
- Much faster than Hadoop MapReduce.

Fast Processing

- Up to 100x faster than Hadoop (in-memory).
- Reduces disk I/O operations.

Ease of Use

Supports multiple languages:

- Scala
- Python
- Java
- R

Lazy Evaluation

Operations are not executed immediately.
Execution happens only when an action is called.

Fault Tolerance

Uses RDD lineage to recover lost data.

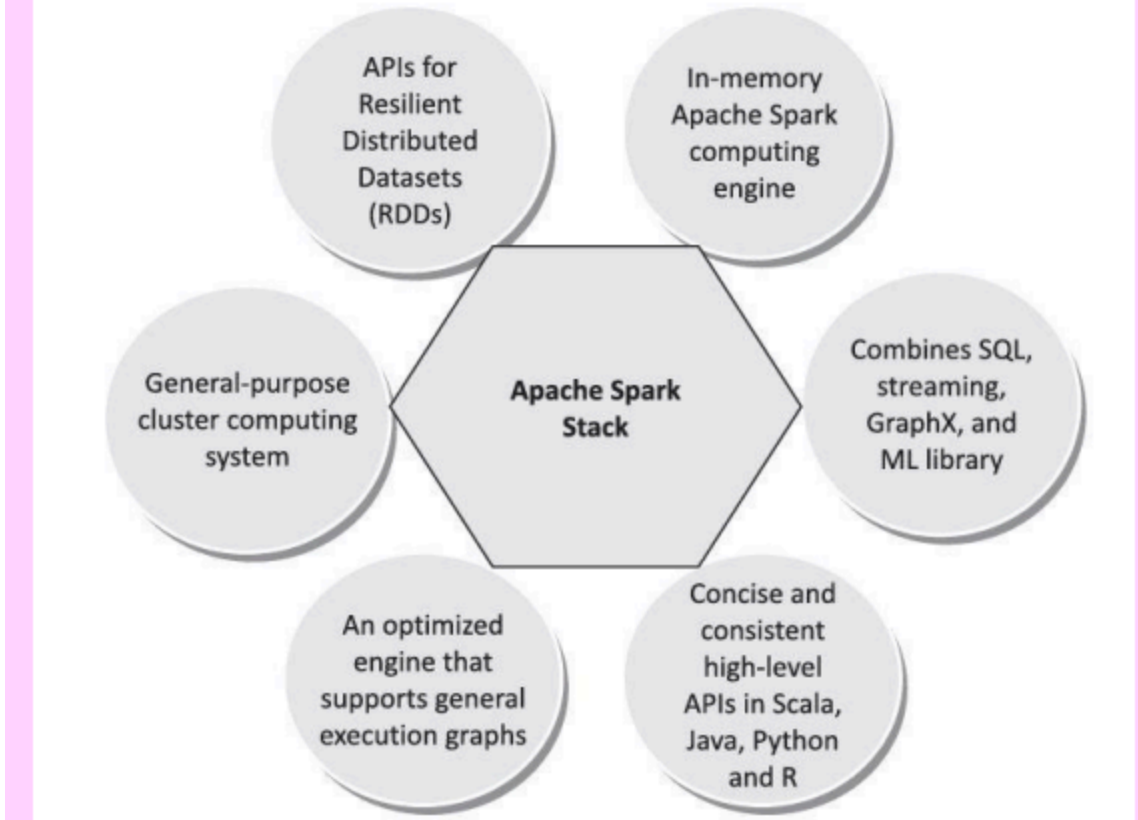
Real-Time Processing

Supports streaming via Spark Streaming.

Machine Learning & Graph Processing

- MLlib for ML algorithms
- GraphX for graph analytics

Figure 5.2 Main features of Spark

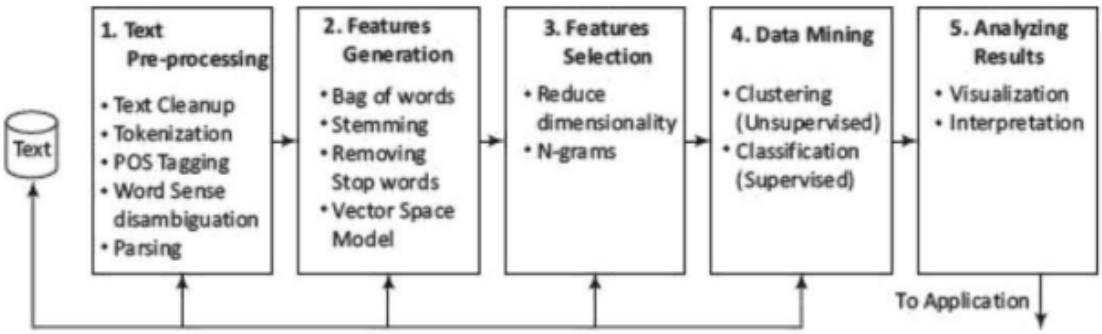


Q.10	a.	Define text mining. With a neat diagram explain the text mining process.	10	L2	CO
	b.	With a neat diagram, explain the three phases of web usage mining.	10	L2	CO

10 (a) Define Text Mining. With a neat diagram explain the text mining process.

Text Mining is a rapidly evolving area of research. As the amount of social media and other

text data grows, there is need for efficient abstraction and categorization of meaningful information from the text.



The five phases for processing text are as follows:

The five phases for processing text are as follows:

Phase 1: Text pre-processing enables Syntactic/Semantic text-analysis and does the followings:

1. Text cleanup is a process of removing unnecessary or unwanted information. Text cleanup converts the raw data by filling up the missing values, identifies and removes outliers, and resolves the inconsistencies. For example, removing comments, removing or escaping "%20" from URL for the web pages or cleanup the typing error, such as teh (the), do n't (do not) [%20 specifies space in a URL].

2. Tokenization is a process of splitting the cleanup text into tokens (words) using white spaces and punctuation marks as delimiters.

3. Part of Speech (POS) tagging is a method that attempts labeling of each token (word) with an appropriate POS. Tagging helps in recognizing names of people, places, organizations and titles. English language set includes the noun, verb, adverb,

adjective,

prepositions and conjunctions. Part of Speech encoded in the annotation system of the Penn

Treebank Project has 36 POS tags.4

4. Word sense disambiguation is a method, which identifies the sense of a word used in

a sentence; that gives meaning in case the word has multiple meanings. The

methods, which resolve the ambiguity of words can be context or proximity based.

Some examples of such words are bear, bank, cell and bass.

5. Parsing is a method, which generates a parse-tree for each sentence. Parsing

attempts and infers the precise grammatical relationships between different words in

a given sentence.

Phase 2: Features Generation is a process which first defines features (variables,

predictors). Some of the ways of feature generations are:

1. Bag of words-Order of words is not that important for certain applications.

Text document is represented by the words it contains (and their occurrences).

Document classification methods commonly use the bag-of-words model. The pre-

processing of a document first provides a document with a bag of words. Document

classification methods then use the occurrence (frequency) of each word as a feature

for training a classifier. Algorithms do not directly apply on the bag of words, but use

the frequencies.

2. Stemming-identifies a word by its root.

(i) Normalizes or unifies variations of the same concept, such as speak for three

variations, i.e., speaking, speaks, speakers denoted by [speaking, speaks, speaker-

+ speak]

(ii)Removes plurals, normalizes verb tenses and removes affixes.

Stemming reduces the word to its most basic element. For example, impurification -+ pure.

3. Removing stop words from the feature space-they are the common words, unlikely to

help text mining. The search program tries to ignore stop words. For example,

ignores a, at, for, it, in

and are.

4. Vector Space Model (VSM)-is an algebraic model for representing text documents as vector

of identifiers, word frequencies or terms in the document index. VSM uses the method of

term frequency-inverse document frequency (TF-IDF) and evaluates how important is a

word in a document.

When used in document classification, VSM also refers to the bag-of-words model. This bag

of words is required to be converted into a term-vector in VSM. The term vector provides

the numeric values corresponding to each term appearing in a document. The term vector is

very helpful in feature generation and selection.

Term frequency and inverse document frequency (IDF) are important metrics in text analysis.

TF-IDF weighting is most common- Instead of the simple TF, IDF is used to weight the

importance of word in the document.

Phase 3: Features Selection is the process that selects a subset of features by rejecting

irrelevant and/or redundant features (variables, predictors or dimension) according to defined criteria. Feature selection process does the following:

1. Dimensionality reduction-Feature selection is one of the methods of division and therefore,

dimension reduction. The basic objective is to eliminate irrelevant and redundant data.

Redundant features are those, which provide no extra information. Irrelevant features

provide no useful or relevant information in any context.

Principal Component Analysis (PCA) and Linear Discriminate Analysis (LDA) are dimension

reduction methods. Discrimination ability of a feature measures relevancy of features.

Correlation helps in finding the redundancy of the feature. Two features are redundant to

each other if their values correlate with each other.

2. N-gram evaluation-finding the number of consecutive words of interest and extract them.

For example, 2-gram is a two words sequence, ["tasty food", "Good one"]. 3-gram is a three

words sequence, ["Crime Investigation Department"].

3. Noise detection and evaluation of outliers methods do the identification of unusual or

suspicious items, events or observations from the data set. This step helps in cleaning the

data.

The feature selection algorithm reduces dimensionality that not only improves the

performance of learning algorithm but also reduces the storage requirement for a dataset. The

process enhances data understanding and its visualization.

Phase 4: Data mining techniques enable insights about the structured database that resulted

from the previous phases. Examples of techniques are:

1. Unsupervised learning (for example, clustering)

(i) The class labels (categories) of training data are unknown

(ii)Establish the existence of groups or clusters in the data

Good clustering methods use high intra-cluster similarity and low inter-cluster similarity.

Examples of uses - biogs, pattern

and trends.

2. Supervised learning (for example, classification)

(i) The training data is labeled indicating the class

(ii)New data is classified based on the training set

Classification is correct when the known label of test sample is identical with the resulting

class computed from the classification model.

Examples of uses are news filtering application, where it is required to automatically assign

incoming documents to pre-defined categories; email spam filtering, where it is identified

whether incoming email messages are spam or not.

Example of text classification methods are Naive Bayes Classifier and SVMs.

3. Identifying evolutionary patterns in temporal text streams-the method is useful in a wide

range of applications, such as summarizing of events in news articles and extracting the

research trends in the scientific literature.

Phase 5: Analysing results

(i) Evaluate the outcome of the complete process.

(ii) Interpretation of Result- If acceptable then results obtained can be used as an input for next

set of sequences. Else, the result can be discarded, and try to understand what and why the

process failed.

(iii) Visualization - Prepare visuals from data, and build a prototype.

(iv)Use the results for further improvement in activities at the enterprise, industry or institution.

10 (b) With a neat diagram, explain the three phases of Web Usage Mining

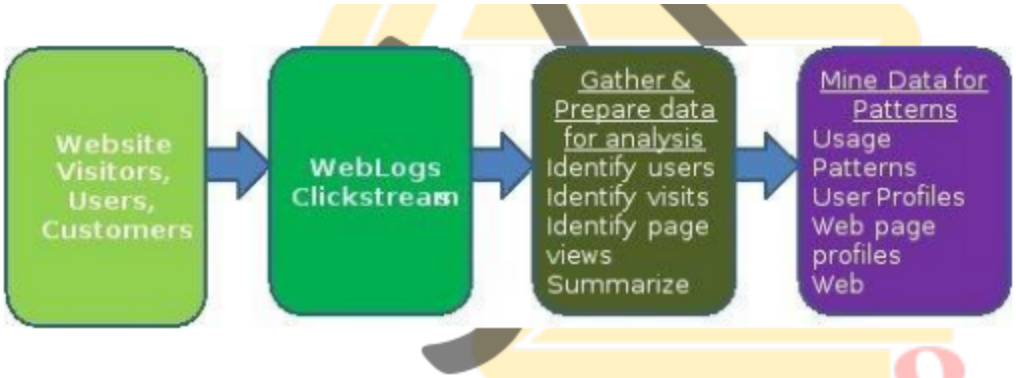


Figure: 2.2 Web Usage Mining architecture

The web content could be analyzed at multiple levels (Figure 2.2).

1. The server side analysis would show the relative popularity of the web pages accessed.

Those websites could be hubs and authorities.

2. The client side analysis could focus on the usage pattern or the actual content consumed and created by users.

1. Usage pattern could be analyzed using 'clickstream' analysis, i.e. analyzing web activity for patterns of sequence of clicks, and the location and duration of visits on websites. Clickstream analysis can be useful for web activity analysis, software testing, market research, and analyzing employee productivity.

2. Textual information accessed on the pages retrieved by users could be analyzed using text mining techniques. The text would be gathered and structured using the bag-of-words technique to build a Term-document matrix. This matrix could then be mined using cluster analysis and association rules for patterns such as popular

Web usage mining has many business applications. It can help predict user behavior based on previously learned rules and users' profiles, and can help determine lifetime value of clients. It can also help design cross-marketing strategies across products, by observing association rules among the pages on the website. Web usage can help evaluate promotional campaigns and see if the users

were attracted to the website and used the pages relevant to the campaign. Web usage mining could be used to present dynamic information to users based on their interests and profiles. This includes targeted online ads and coupons at user groups based on user access patterns.