AKTU B.Tech 3rd Year

CS & CS Allied

Data Analytics

Unit-5

in ONE SHOT



AKTU PYQs Covered

By Shubhangi Ma'am

GATEWAY CLASSES

SYLLABUS

UNIT V

Frameworks and Visualization

MapReduce, Hadoop, Pig, Hive, HBase, MapR, Sharding, NoSQL Databases, S3, Hadoop Distributed File Systems, Visualization: visual data analysis techniques, interaction techniques, systems and applications.

Introduction to R : R graphical user interfaces, data import and export, attribute and data types, descriptive statistics, exploratory data analysis, visualization before analysis, analytics for unstructured data.

GATEWAY CLASSES

PREVIOUS YEAR IMPORTANT QUESTIONS

Q.1 : Differentiate between Pig & SQL. **AKTU(2023-24)**

Q.2 : Discuss Various Visualization techniques with Suitable examples . **AKTU(2023-24)**

Q.3 : Draw & Discuss the architecture of HDFS in detail. **AKTU(2023-24)**

Q.4 : Define Shard & Database shard. **AKTU(2022-23)**

Q.5 : Discuss Various Classification of Visualization Techniques . **AKTU(2022-23)**

Q.6 : What is Exploratory Data Analytics(EDA)?

Q.7 : Explain the working of MapReduce with intermediate Key – value pairs. **AKTU(2025-26)**

Q.8 : Design a Hadoop – based framework for storing & processing large – scale data. **AKTU(2025-26)**

Q.9 : Design an R – based workflow for analyzing unstructured text data . **AKTU(2025-26)**

Q.10 : Draw the Architecture of HIVE its features. **AKTU(2024-25)**

Q.11 : Explain the role of Hive in the Hadoop ecosystem. **AKTU(2023-24)**

GATEWAY CLASSES

GATEWAY CLASSES

Q.12 : How does the MapReduce framework facilitate distributed processing? **AKTU(2023-24)**

Q.13 : Discuss the concept of sharding in the context of NoSQL databases . How does sharding contribute to Scalability , what challenges does it address? **AKTU(2023-24)**

Q.14 : Differentiate between Pig & SQL . **AKTU(2023-24)**

Q.15 : Discuss various visualization techniques with suitable examples . **AKTU(2023-24)**

Q.16 : Draw and discuss the architecture of HDFs in detail. **AKTU(2023-24)**

Q.17 : What are the approaches to integrate the human in data exploration process to realize different types of approaches to visual data mining? **AKTU(2023-24)**

Q.18 : Define HDFS. Discuss the HDFS architecture & HDFS commands in brief . **AKTU(2023-24)**

Q.19 : Differentiate between Map Reduce & Apache Pig . **AKTU(2023-24)**

MAPREDUCE

MapReduce is a **distributed data processing model** used to process and analyze large datasets efficiently across multiple computers.

It divides a large processing task into two main phases:

- **Map Phase –** converts input data into intermediate key-value pairs.
- **Reduce Phase –** processes the grouped key-value pairs to produce the final result.

The complete process is :

Input → Map → Shuffle & Sort → Reduce → Output

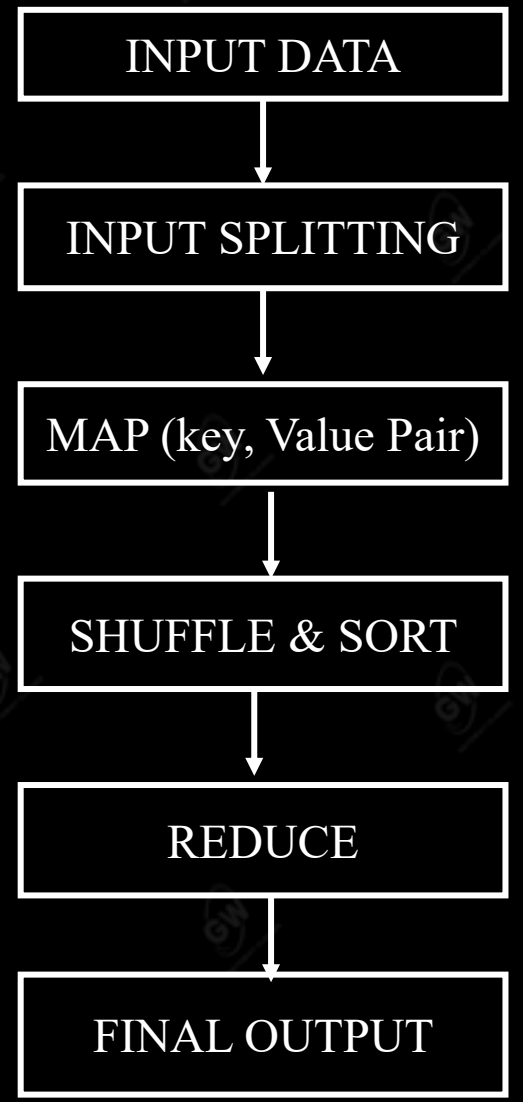
WORKING OF MAP REDUCE

1. Input Splitting

- The input data is divided into smaller blocks before processing starts.
- Each block is assigned to different nodes in the cluster.
- This division helps in parallel and faster processing.

2. Map Phase

- In this phase, input data is processed and converted into key-value pairs.
- Each mapper works independently on a data block.
- It is the first step of actual data processing.



GATEWAY CLASSES

GATEWAY CLASSES

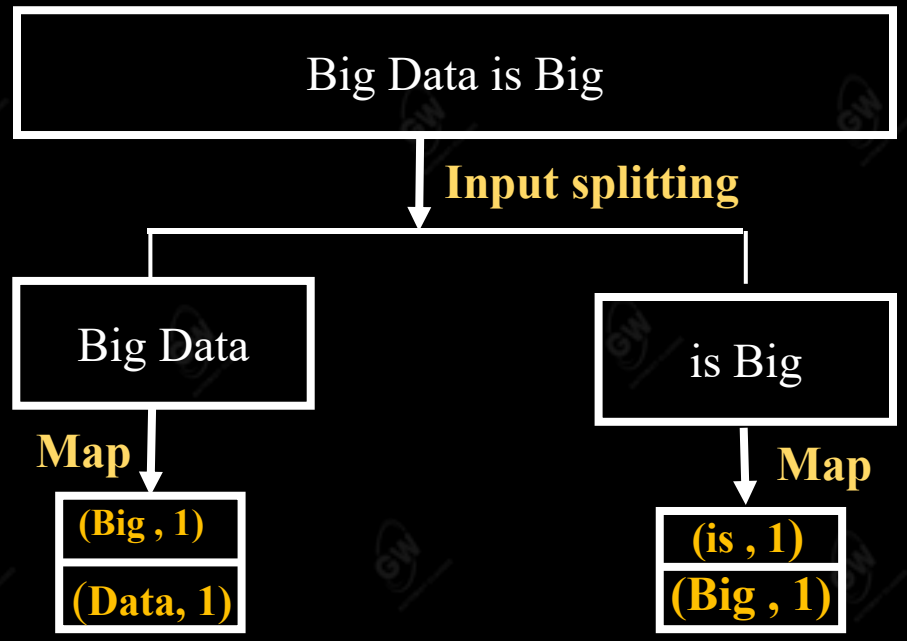
3. Shuffle and Sort

- The system groups all similar keys together automatically.
- Data is sorted so that reducer gets organized input .
- This phase acts as a bridge between Map and Reduce.

4. Reduce Phase

- The reducer takes grouped key-value pairs as input.
- It combines values of the same key to produce final output .
- This is the final step where meaningful result is generated.

GW GATEWAY CLASSES



1. Input
2. Split
3. Map
4. Shuffle
5. Reduce
6. Output

Shuffle & Sort (Grouping)

Big - [1, 1]
Data - [1]
I s - [1]

Reduce

Big -2
Data – 1
is -1

→ (Big, 2) (Data,1) (is, 1)

GATEWAY CLASSES

ADVANTAGES & LIMITATIONS

GATEWAY CLASSES

Advantages of MapReduce

- Parallel processing – Processes data on multiple machines.
- Scalable – Handles very large datasets.
- Fault tolerant – Can handle machine failures.
- Cost effective – Works on commodity hardware.

Limitations of MapReduce

- Slow for real-time processing.
- High disk I/O due to intermediate data.
- Not suitable for iterative tasks.
- Shuffle and sort adds overhead.

HADOOP

Hadoop is an open-source distributed framework used to store and process large amounts of Big Data across multiple computers.

Instead of storing and processing all data on a single computer, Hadoop distributes the data and processing tasks among multiple machines called nodes.

Need for Hadoop

- When the amount of data is very large
- A single computer may not be able to store the entire data.
- Processing on one machine can be slow.
- Failure of one machine can cause data loss.
- Hadoop solves these problems by providing distributed storage, parallel processing, scalability, and fault tolerance.

GATEWAY CLASSES

MAIN COMPONENTS OF HADOOP

GATEWAY CLASSES

1. HDFS (Hadoop Distributed File System)

- HDFS is used to store large amounts of data across multiple machines.
- Large files are divided into smaller blocks.
- These blocks are stored on different nodes.
- Multiple copies of data can be maintained for fault tolerance.

2. MapReduce

- MapReduce is used to process large datasets in a distributed manner.
- It mainly works in two phases:
- Map: Processes the input data and produces key-value pairs.
- Reduce: Combines the intermediate results to produce the final output.

GATEWAY CLASSES

3. YARN

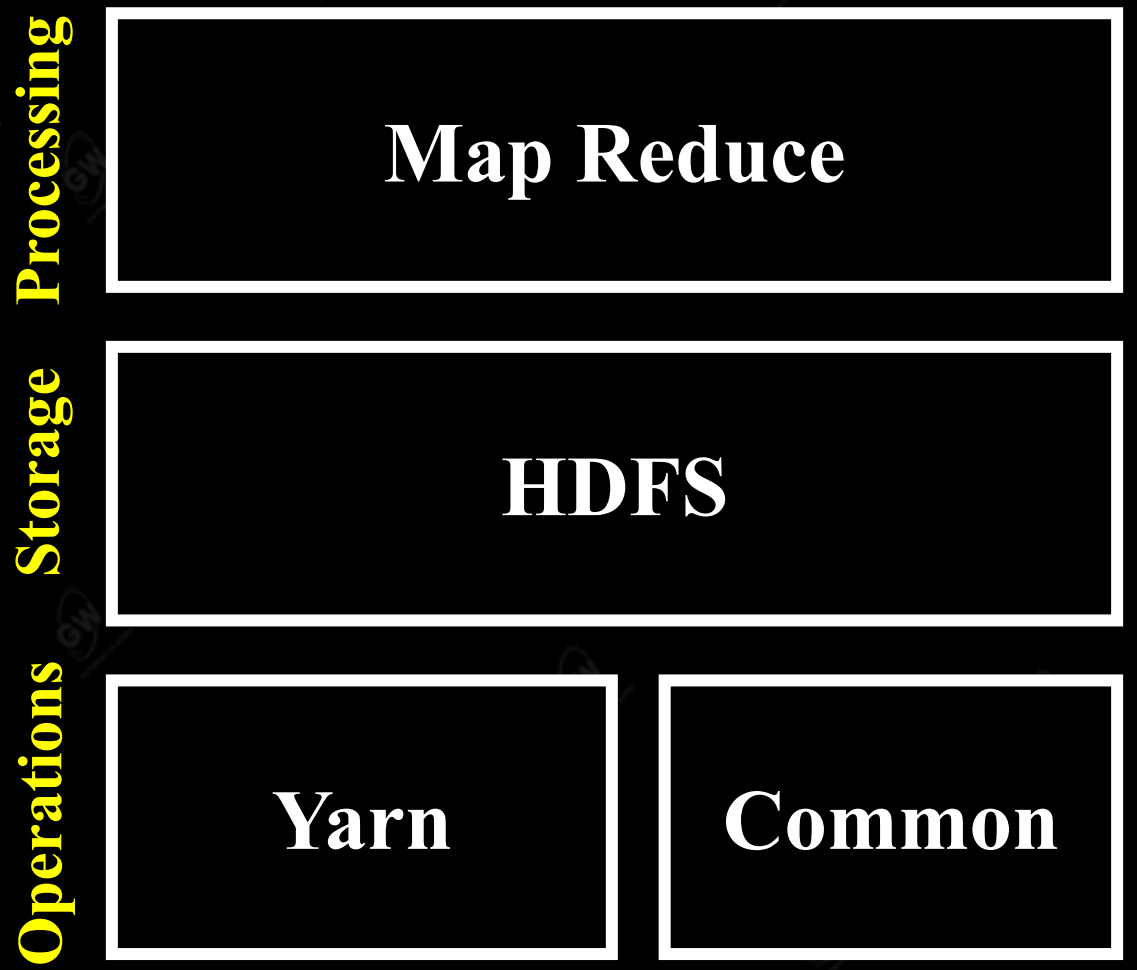
YARN (Yet Another Resource Negotiator) is responsible for resource management and job scheduling in a Hadoop cluster.

It manages :

- CPU and memory resources
- Execution of different jobs
- Allocation of resources to applications

4. Hadoop Common

Hadoop Common provides the common libraries and utilities required by other Hadoop components.



The Four Components of Hadoop

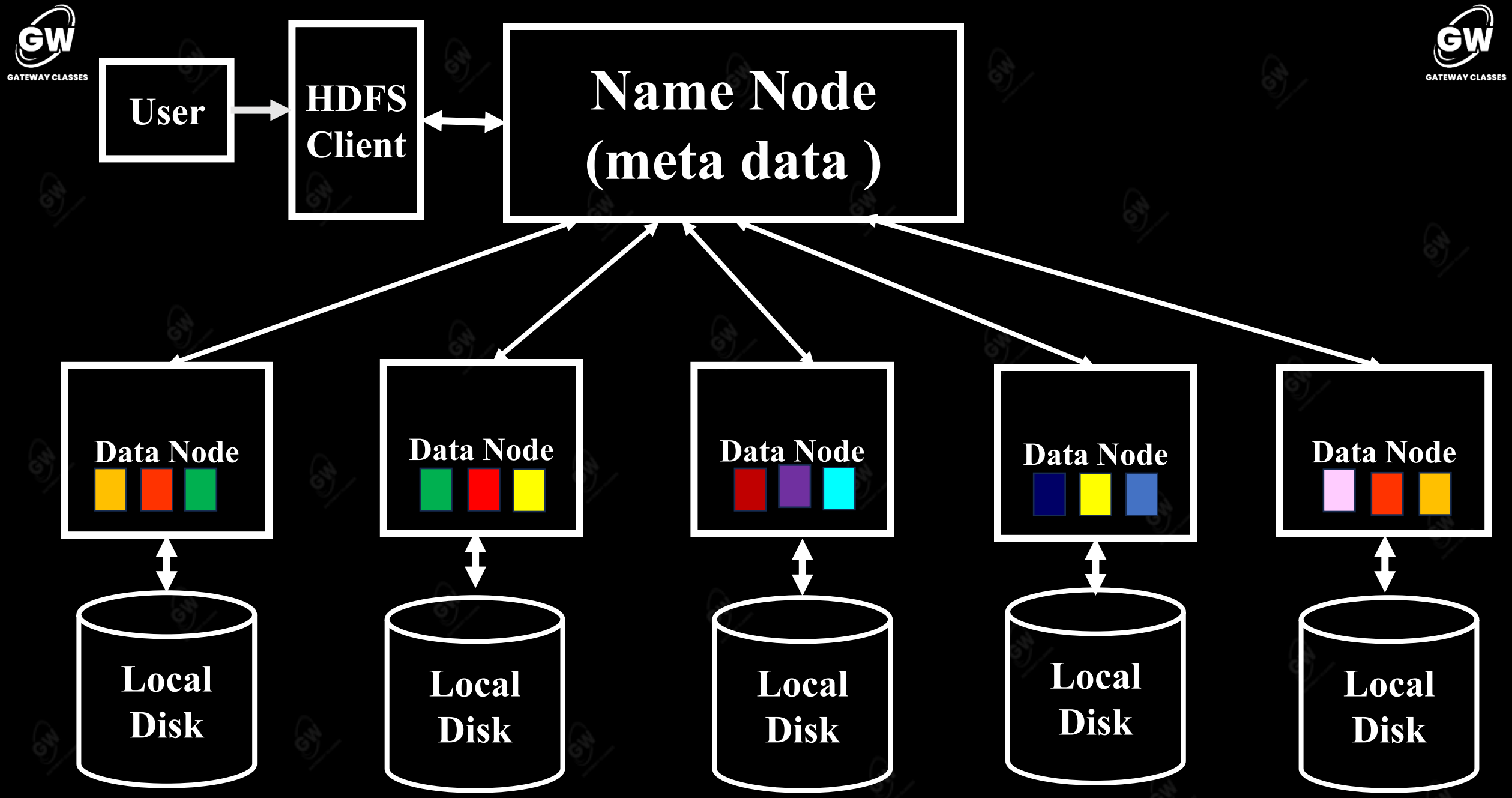
GATEWAY CLASSES

HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

GATEWAY CLASSES

HDFS (Hadoop Distributed File System) is a storage system used in **Apache Hadoop** to store large amounts of data across multiple computers.

It divides big files into smaller blocks and stores them on different machines in a cluster. It also keeps multiple copies of data to ensure safety and reliability, even if some machines fail.


GATEWAY CLASSES
User
HDFS Client
Name Node (meta data)
Data Node
Data Node
Data Node
Data Node
Data Node
Local Disk
Local Disk
Local Disk
Local Disk
Local Disk

GATEWAY CLASSES

User

↓

Request to **write data**

↓

HDFS Client contacts NameNode

↓

NameNode checks metadata and decides DataNode locations

↓

HDFS Client gets DataNode addresses

↓

Data is divided into blocks

↓

Blocks are stored in DataNodes with replication

↓

Write operation performed

GATEWAY CLASSES

User

↓

Request to **read data**

↓

HDFS Client contacts NameNode

↓

NameNode checks metadata and provides DataNode locations

↓

HDFS Client gets nearest DataNode address

↓

Client reads blocks from DataNodes

↓

Blocks are combined

↓

File returned to user

WORKING OF HDFS

1. File Input

User uploads a file into HDFS. The system does not store it as one single file. Instead, it prepares the file to be divided into smaller parts. This helps in handling large data easily.

2. Block Division

HDFS splits the file into small blocks (usually 128 MB). Each block is treated as a separate unit. These blocks can be stored on different machines. This improves speed and efficiency.

3. Replication

Each block is copied multiple times (default is 3). These copies are stored on different machines. This ensures data safety if any system fails, Replication provides fault tolerance.

4.Storage in DataNodes

DataNodes are worker machines that store actual data blocks. They manage reading and writing of data. Each DataNode stores multiple blocks. They work under the control of the NameNode.

GW
GATEWAY CLASSES
GW
GATEWAY CLASSES

5. NameNode (Metadata)

NameNode is the master node in HDFS. It stores metadata like file names and block locations. It does not store actual data. It controls and manages the whole system.

6. Read Operation

When a client wants to read a file, it contacts the NameNode. The NameNode provides the locations of blocks. The client then reads data directly from DataNodes. Finally, blocks are combined to form the file.

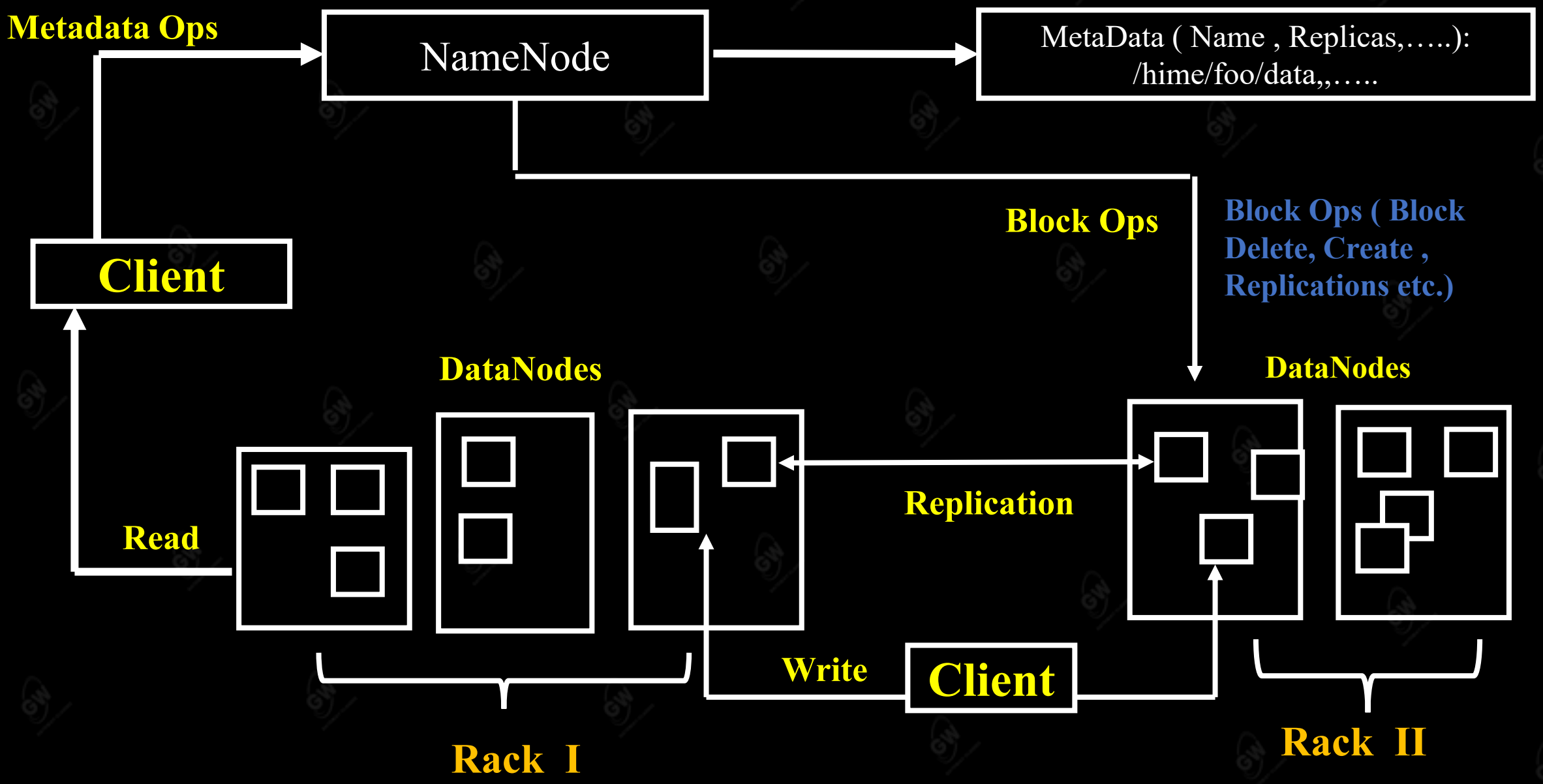
7. Write Operation

When writing a file, the client first contacts the NameNode. The NameNode selects suitable DataNodes. The client sends data blocks to those nodes. Replication of blocks happens automatically during this process.

8. Fault Tolerance

If any DataNode fails, data is still safe due to replication. The system detects the failure automatically. It creates new copies from existing replicas. This ensures continuous data availability.

COMPONENTS OF HDFS / ARCHITECTURE OF HDFS



GATEWAY CLASSES

GATEWAY CLASSES

1. Client

Client is the user or application that interacts with HDFS. It sends requests to store (write) or access (read) data. The client does not directly manage storage. It first contacts the NameNode for instructions.

2. NameNode (Master Node)

NameNode is the brain of HDFS. It manages all metadata like file names, block details, and locations. It decides where data should be stored. It never stores actual data, only information about data.

3. Metadata (Shown on Right Side)

Metadata means information about the file, not the file itself. It includes file path (like /home/foo/data), number of blocks, and replicas. This data is stored in the NameNode. It helps in locating actual data quickly.

GATEWAY CLASSES

GATEWAY CLASSES

4. Metadata Operations (Top Left Arrow)

These are requests between Client and NameNode. The client asks for file details, permissions, or block locations. NameNode responds with required information. No actual data transfer happens here.

5. DataNodes (Worker Nodes)

DataNodes are machines that store the actual data blocks. Each DataNode holds many blocks of different files.

They perform read and write operations. They also send regular updates (heartbeats) to the NameNode.

6. Block Operations (Middle Arrow)

Block operations happen between NameNode and DataNodes. The NameNode instructs DataNodes to store, delete, or replicate blocks. These operations control how data is managed internally. It ensures proper distribution of data.

GATEWAY CLASSES
GATEWAY CLASSES

7. Replication (Center)

Replication means creating multiple copies of each block. These copies are stored on different DataNodes. It increases fault tolerance and data safety. Even if one node fails, data is still available.

8. Racks (Rack I & Rack II)

Racks represent groups of machines in a data center. Data is distributed across different racks. This improves reliability because even if one rack fails, data can be accessed from another. It is called rack awareness.

9. Write Operation (Bottom Middle)

When writing data, the client first contacts the NameNode. The NameNode selects DataNodes for storing blocks. The client sends data to one DataNode, which then forwards it to others for replication. This forms a pipeline.

GW GATEWAY CLASSES

10. Read Operation (Left Bottom)

GW GATEWAY CLASSES

For reading, the client asks the NameNode for block locations. The NameNode replies with nearest DataNodes. The client directly reads data from those DataNodes. This makes reading fast and efficient.

FEATURES OF HDFS

Feature	Meaning
Distributed Storage	HDFS stores large data across multiple machines instead of a single system.
Fault Tolerance	It keeps multiple copies (replication) of data, so data is safe even if a machine fails.
High Scalability	New machines can be easily added to handle more data.
High Throughput	It provides fast access and processing of large datasets.
Data Replication	Each data block is copied on different nodes for reliability.
Large Data Handling	It is specially designed to store and manage very big files.

GW
GATEWAY CLASSES

ROLE OF HADOOP IN BIG DATA

GW
GATEWAY CLASSES

1. Storage of Big Data : Hadoop uses **HDFS** to store very large datasets across multiple machines.

2. Processing of Big Data : Hadoop uses **MapReduce** to process large datasets in parallel.

3. Scalability : Hadoop allows additional machines to be added to the cluster as data grows.

4. Fault Tolerance : Hadoop keeps multiple copies of data so that data can still be accessed if a machine fails.

5. Distributed Processing : Instead of processing all data on a single machine, Hadoop distributes the work among multiple machines.

6. Handling Different Types of Data : Hadoop can work with large volumes of **structured, semi-structured, and unstructured data**.

PIG

- **Apache Pig** is a **high-level data flow platform** used to process and analyze large datasets.
- It provides a simple way to perform data-processing operations without writing complex MapReduce programs directly.
- Pig is mainly used with **Hadoop** for processing large amounts of data.

Key Points

- It is designed for **Big Data processing**.
- It works on top of the Hadoop ecosystem.
- It simplifies complex data-processing tasks.
- It can process large datasets stored in Hadoop.

COMPONENTS OF APACHE PIG

Pig Latin : Pig Latin is the **high-level scripting language** used by Pig. It provides simple commands to describe how data should be :

- Loaded
- Filtered
- Grouped
- Sorted
- Joined
- Stored

Example :

```
A = LOAD 'student.txt';
B = FILTER A BY marks > 60;
DUMP B;
```

Use of Pig in Data Processing

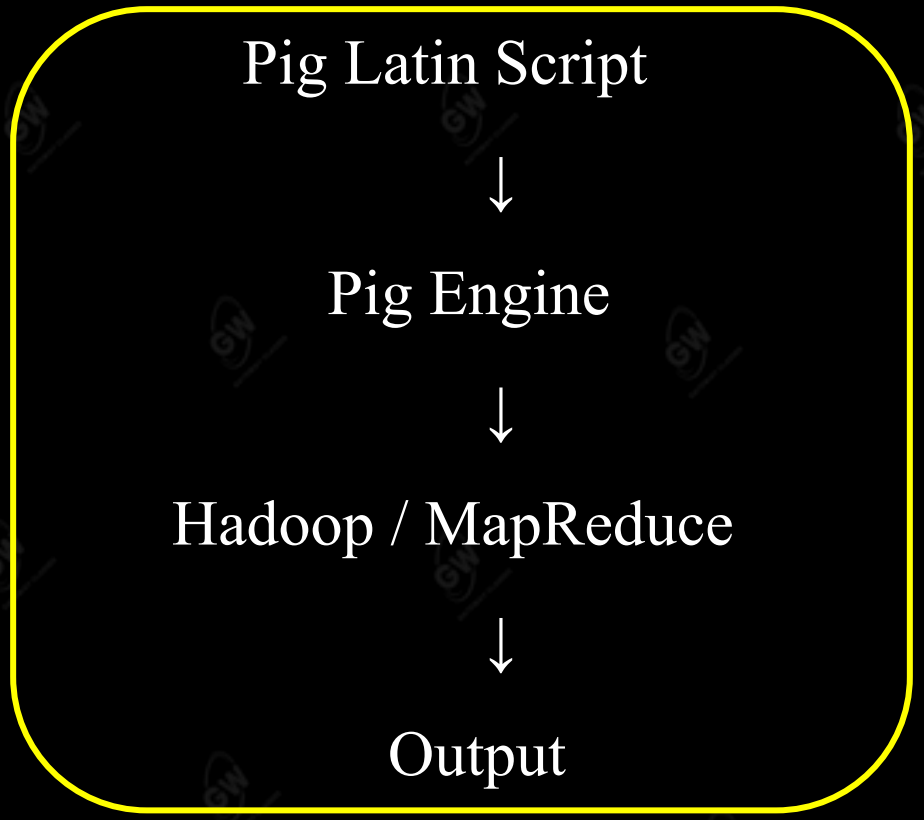
1. **Loading large datasets**
2. **Filtering unwanted data**
3. **Grouping data**
4. **Sorting data**
5. **Joining datasets**
6. **Transforming data**
7. **Storing processed results**

GATEWAY CLASSES

2. Pig Engine : Pig Engine is responsible for executing the Pig Latin scripts.

It converts Pig Latin statements into execution jobs, which can run on Hadoop.

It coordinates the processing of data and produces the required output.



GATEWAY CLASSES

GW
GATEWAY CLASSES

PIG OPERATIONS

GW
GATEWAY CLASSES

OPERATION	DESCRIPTION
LOAD	Loads data from a file or external source into Pig.
STORE	Saves the processed data into a file or storage location.
DUMP	Displays the contents of a relation on the screen.
FILTER	Selects records that satisfy a given condition.
LIMIT	Restricts the number of records in the output.
UNION	Combines two or more relations into a single relation.
SPLIT	Divides a relation into two or more relations based on conditions.
DISTINCT	Removes duplicate records from a relation.

GATEWAY CLASSES

ADVANTAGES / LIMITATIONS

GATEWAY CLASSES

Advantages of Pig

- Easy to use – Pig Latin is simpler than MapReduce.
- Handles Big Data efficiently.
- Less coding required.
- Supports complex data processing operations.

Limitations of Pig

- Slower than some modern processing tools.
- Not suitable for real-time processing.
- Limited for complex analytics.
- Mainly used with the Hadoop ecosystem.

HIVE

- **Apache Hive** is a **data warehouse system** built on Hadoop that is used to store, manage, and analyze large datasets.
- It provides a **SQL-like language** called **HiveQL**, which makes it easier for users familiar with SQL to work with Big Data.
- Hive is mainly designed for **data analysis and querying**, rather than real-time transaction processing.

Features of Hive

- SQL-like language: Uses HiveQL, which is similar to SQL.
- Scalable: Works with distributed Hadoop systems.
- Data Warehousing: Useful for storing and analyzing large datasets.
- Supports different data formats: Can work with various types of data.
- Query processing: Supports operations such as filtering, grouping, joining and sorting.
- Integration with Hadoop: Works with HDFS and Hadoop processing tools.

ARCHITECTURE / COMPONENTS OF APACHE HIVE

1. HiveQL (Hive Query Language)

- HiveQL is a **SQL-like query language** used to interact with data in Hive.
- It is used for **querying, filtering, grouping, sorting, and analyzing** large datasets.
- It is easier to use for users familiar with SQL.

2. Driver

- The **Driver** receives queries submitted by the user.
- It manages the **complete lifecycle of a query**, from submission to result.
- It coordinates the different stages of query execution.

3. Compiler

- The **Compiler** analyzes the HiveQL query.
- It checks the query for **syntax and semantic errors**.
- It converts the query into an **execution plan**.

4. Metastore

- The **Metastore** stores **metadata** about the data.
- It contains information such as : Table names , Column names , Data types , Table location ,
- It helps Hive understand the structure and location of data.

5. Execution Engine

- The **Execution Engine** executes the execution plan created by the compiler.
- It coordinates the processing of the query using the underlying Hadoop processing framework.
- It returns the **final result** to the user.

Working of Hive

- User submits a HiveQL query.
- Driver receives the query.
- Compiler checks and converts it into an execution plan.
- Metastore provides required metadata.
- Execution Engine executes the plan using Hadoop.
- Data is read from HDFS.
- The result is returned to the user.

GW
GATEWAY CLASSES

DIFFERENCE BETWEEN HIVE AND PIG

GW
GATEWAY CLASSES

Basis	Hive	Pig
Definition	A data warehouse tool used to query and analyze large datasets.	A data flow platform used to process and transform large datasets.
Language	Uses HiveQL (HQL), which is SQL-like.	Uses Pig Latin scripting language.
Approach	SQL-based approach.	Data-flow based approach.
Main Use	Mainly used for data querying and analysis.	Mainly used for data processing and transformation.
Ease of Use	Easier for users familiar with SQL.	Easier for users working with data-flow operations.

GW
GATEWAY CLASSES

HBASE

GW
GATEWAY CLASSES

HBase is a **distributed, open-source NoSQL database** designed to store and manage very large amounts of data.

It is built on top of **Hadoop HDFS** and is suitable for handling **large, sparse datasets**.

Unlike traditional relational databases, HBase stores data in a **column-oriented structure** and can provide fast access to individual records.

Need for HBase

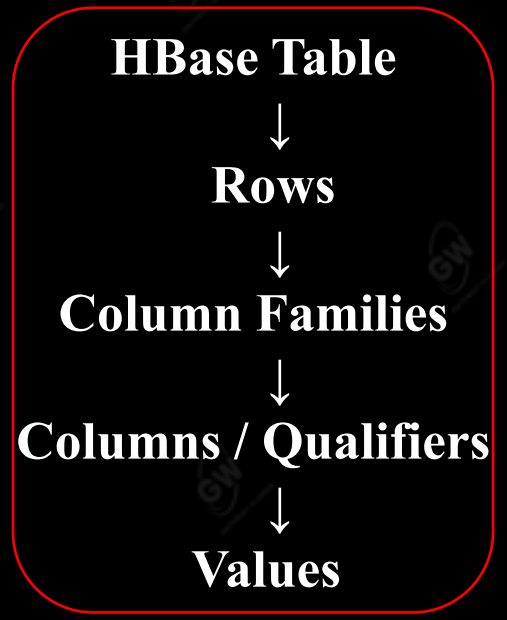
- HDFS is mainly used for storing large files, but it is not designed for fast random access to individual records.
- HBase provides **quick access to specific rows or records**.
- It is suitable for **large and continuously growing datasets**.

BASIC STRUCTURE OF HBASE

HBase stores data in the form of **tables**.

Main Elements

- **Table:** Collection of related data.
- **Row:** A single record in the table.
- **Row Key:** Unique identifier of a row; used to quickly locate a record.
- **Column Family:** Group of related columns.
- **Column/Qualifier:** Individual attribute within a column family.
- **Value:** Actual data stored for a column.



Example :

Row Key	Name	Age	Course
101	Aman	20	B.Tech
102	Riya	21	BCA

Here, **101, 102, 103 are Row Keys**.

HBASE ARCHITECTURE

HBase mainly works with the following components:

1. HMaster

- Manages the HBase cluster.
- Handles administrative tasks such as assigning and managing regions.

2. RegionServer

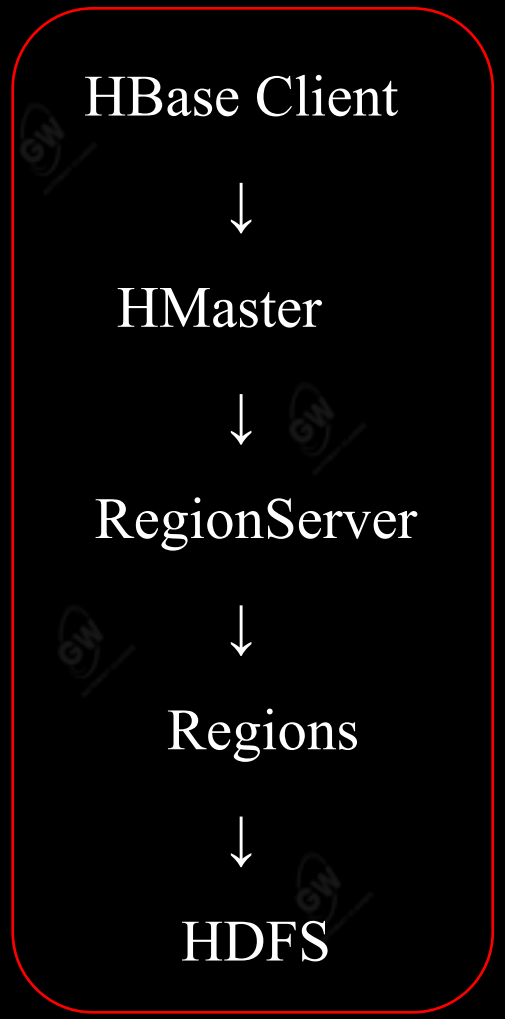
- Stores and manages actual data.
- Handles **read and write requests** from clients.

3. Region

- A table is divided into smaller parts called **regions**.
- Each RegionServer manages one or more regions.

4. HDFS

- Provides the underlying **distributed storage** for HBase data.



GATEWAY CLASSES

BASIC WORKING OF HBASE

GATEWAY CLASSES

1. Data is stored in an **HBase table**.
2. Large tables are divided into **regions**.
3. Regions are managed by **RegionServers**.
4. When a client requests data, the appropriate RegionServer is contacted.
5. Data is read from or written to the underlying **HDFS** storage.

USE OF HBASE IN BIG DATA

- Storing very large datasets.
- Handling **sparse data** efficiently.
- Providing fast read/write access.
- Managing data that continuously grows.
- Supporting applications that need random access to large datasets.
- Working with data stored in the Hadoop environment.

MAPR

MapR was a **commercial Big Data platform/distribution** based on the Hadoop ecosystem.

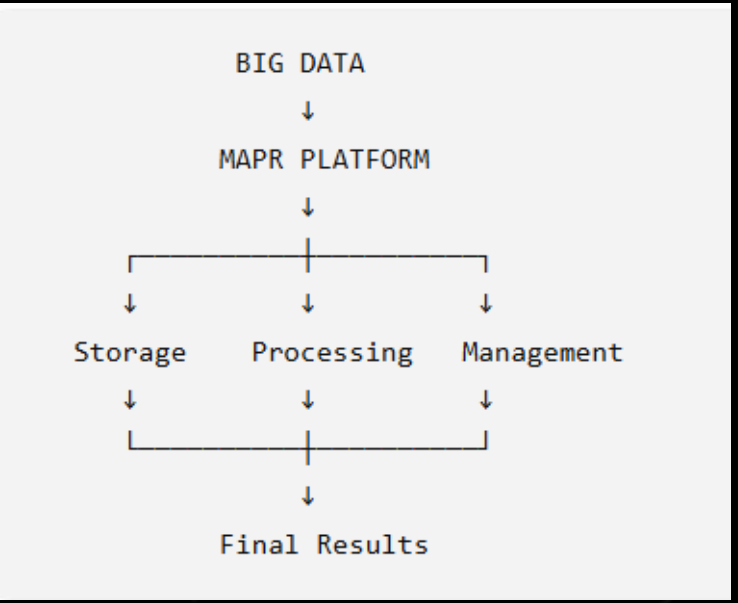
It provided technologies for **storing, processing, and managing large amounts of data** in a distributed environment.

MapR was designed to provide an enterprise platform for Big Data applications.

Role of MapR as a Big Data Platform

MapR provided a platform for managing and processing Big Data across a cluster of machines.

- **Distributed data storage**
- **Large-scale data processing**
- **Cluster management**
- Support for different types of Big Data workloads
- Integration with Big Data technologies



SHARDING

Sharding is a technique of **dividing a large dataset into smaller parts called shards** and storing these shards on different servers or machines.

Each shard contains a portion of the complete dataset.

Example : Suppose a database contains **1 million customer records**. Instead of storing all records on one server

- **Shard 1:** Customer 1–333,333
- **Shard 2:** Customer 333,334–666,666
- **Shard 3:** Customer 666,667–1,000,000

The complete dataset is divided using a **sharding key**.

A sharding key is an attribute used to decide **which shard should store a particular record**.

For example, records can be divided based on : Customer ID , User ID , Region , Date

GATEWAY CLASSES
GATEWAY CLASSES

Features

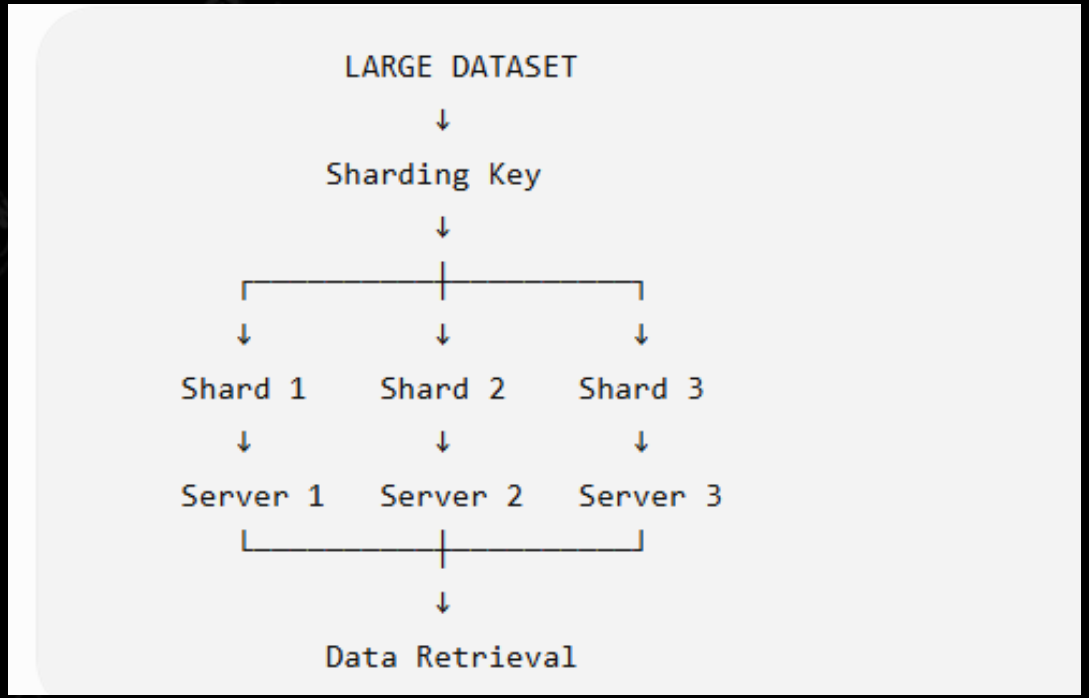
- **Distributed Storage :** Stores data across multiple machines.
- **Scalability :** Can handle increasing amounts of data.
- **High Availability :** Designed to keep services available in distributed environments.
- **Data Processing :** Supports large-scale data processing.
- **Enterprise Support :** Designed for enterprise Big Data applications.

Uses

- Big Data storage
- Data processing
- Real-time and analytical workloads
- Large-scale enterprise applications

WORKING

- Select a suitable **sharding key**.
- Divide the dataset according to that key.
- Store each portion on a different server.
- When a request arrives, the system identifies the appropriate shard.
- The required data is retrieved from that shard.



GATEWAY CLASSES

Advantages of Sharding

- **Scalability :** More servers can be added as data grows.
- **Better Performance :** Workload is distributed across servers.
- **Load Distribution :** No single server has to handle the entire database.
- **Large Data Handling :** Suitable for very large datasets.

Limitations of Sharding

- Complex management – Managing multiple shards is difficult.
- Data imbalance – One shard may receive more data or requests than others.
- Cross-shard queries – Queries involving multiple shards can be slower.
- Resharding difficulty – Changing the shard structure can be complex.
- Sharding key dependency – Poor choice of sharding key can reduce performance.

GATEWAY CLASSES

NO SQL DATABASES

GATEWAY CLASSES

- **NoSQL** stands for **"Not Only SQL."**

  It refers to databases that do not depend only on the traditional relational table structure.
- NoSQL databases are designed to store and manage **large amounts of structured, semi-structured, and unstructured data**.

Main Features

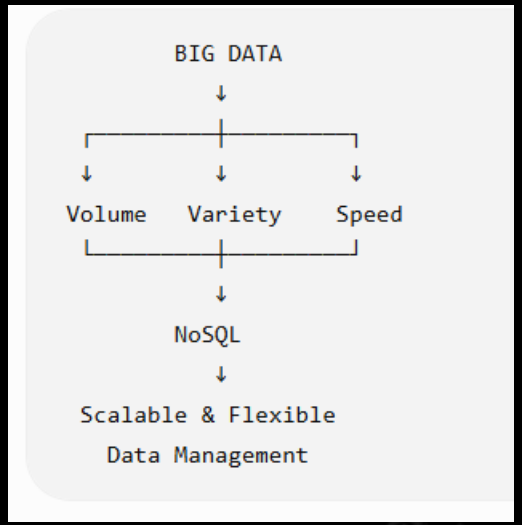
- Handles large volumes of data.
- Supports flexible data structures.
- Can scale across multiple servers.
- Suitable for distributed and Big Data applications.
- Provides fast data access for many types of workloads.

NEED FOR NOSQL

Traditional relational databases may face limitations when data becomes **very large, diverse, and rapidly changing**.

NoSQL is needed because:

- **Large Data Volume :** Handles huge amounts of data.
- **Variety of Data :** Supports structured, semi-structured, and unstructured data.
- **Flexible Schema :** Data structure can change without redesigning fixed tables.
- **Scalability :** Can scale by adding more servers.
- **High Speed :** Provides fast read/write operations for suitable workloads.



TYPES OF NOSQL DATABASES

1. Key-Value Database

Stores data as **key-value pairs**.

Example: Redis

Key → Value

101 → "Aman"

102 → "Riya"

2. Document Database

Stores data in the form of **documents**, commonly JSON-like structures.

Example: MongoDB

```
{
  "name": "Aman",
  "age": 20   }
```

GW
GATEWAY CLASSES

GW
GATEWAY CLASSES

3. Column-Family Database

Stores data in **column families** and is suitable for very large distributed datasets.

Examples: HBase, Cassandra

4. Graph Database

Stores data as **nodes and relationships/edges**.

Example: Neo4j

Aman → FRIEND_OF → Riya

ADVANTAGES OF NOSQL

- **High scalability**
- **Flexible schema**
- Handles **large datasets**
- Supports **distributed storage**
- Fast read/write operations for suitable workloads
- Suitable for **unstructured and semi-structured data**

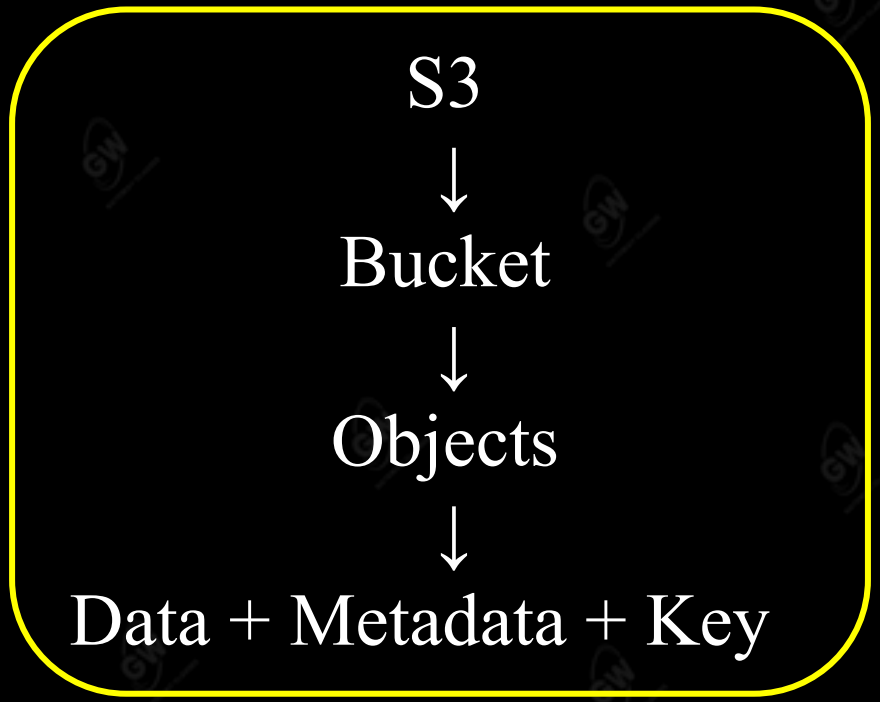
S3(AMAZON SIMPLE STORAGE SERVICE)

S3 Introduction

- **Amazon S3** is a cloud-based **object storage service** provided by AWS.
- It is used to **store and retrieve large amounts of data**.
- It is highly **scalable and durable**.

Object Storage

- S3 stores data as **objects**.
- Objects are stored inside **buckets**.
- Each object has **data, metadata, and a unique key**.



GATEWAY CLASSES

GATEWAY CLASSES

Data Storage & Retrieval

- **Storage :** Data is uploaded as an object into an S3 bucket.
- **Retrieval :** The required object is accessed using its **key**.

Upload → S3 Bucket → Object

↓

Retrieve Data

Use of S3 in Big Data

- Stores **large amounts of Big Data**.
- Used for **data lakes**.
- Stores logs, images, videos, and datasets.
- Provides data storage for **Big Data analytics**.

DATA VISUALIZATION

Data Visualization is the process of representing data in a **visual form using charts, graphs, and plots** to make the data easier to understand and analyze.

Importance of Data Visualization

- Makes large and complex data easier to understand.
- Helps in comparing different values or categories.
- Helps identify patterns and trends.
- Helps identify relationships between variables.
- Helps detect unusual values or outliers.
- Supports data analysis and decision-making.

DATA VISUALIZATION TECHNIQUES

GATEWAY CLASSES

1. Bar Chart

A **Bar Chart** represents data using rectangular bars. The length or height of each bar represents the value of a category.

Example :

A bar chart can be used to compare the marks of different students.

Uses

- Comparing categories.
- Comparing product sales.
- Comparing students' marks.
- Comparing population of different cities.

Key Point:

Bar Chart is mainly used for comparison between categories.

GATEWAY CLASSES
GATEWAY CLASSES

2. Line Chart

A **Line Chart** represents data using points connected by lines. It is mainly used to show **changes and trends over time**.

Example

Monthly sales can be represented using a line chart to show whether sales are increasing or decreasing.

Uses

- Monthly or yearly sales.
- Temperature changes.
- Population growth.
- Performance over time.

Key Point:

Line Chart is mainly used to show trends and changes over time.

GATEWAY CLASSES

3. Pie Chart

A **Pie Chart** is a circular chart divided into different sectors or slices. Each slice represents a **part or percentage of the whole**.

Example : A company's total budget can be divided into:

- 40% for salaries , 30% for marketing , 30% for other expenses

Uses

- Showing percentage distribution.
- Budget distribution.
- Market share.
- Survey results.

Key Point:

Pie Chart is used to represent parts or percentages of a whole.

GW
GATEWAY CLASSES
GW
GATEWAY CLASSES

4. Histogram

A **Histogram** is a graph used to represent the **distribution of numerical data** by dividing the data into intervals or ranges called **bins**.

Example : Students' marks can be divided into:

0–20 || 21–40 || 41–60 || 61–80 || 81–100 The histogram shows how many students fall into each range.

Uses :

- Understanding data distribution.
- Analyzing marks, age, income, height, etc.
- Identifying where most values are concentrated.
- Studying the spread of numerical data.

Key Point :

Histogram is used to show the distribution of numerical data.

GATEWAY CLASSES

GATEWAY CLASSES

5. Scatter Plot

A **Scatter Plot** represents data using individual points. It is used to show the **relationship between two numerical variables**.

Example : To study the relationship between **study hours and marks**, each student's study hours and marks are represented as a point.

Uses

- Study hours vs. marks.
- Height vs. weight.
- Advertising cost vs. sales.
- Temperature vs. electricity consumption.

Key Point :

Scatter Plot is used to identify relationships between two numerical variables.

INTERACTION TECHNIQUES

1. Zooming : **Zooming** is used to increase or decrease the size of a visualization to view data more closely or broadly.

Example : Zooming into a graph to examine a particular area or set of data points.

2. Panning : **Panning** allows the user to move the visualization in different directions, such as left, right, up, or down.

Example : After zooming into a map, panning can be used to view another area.

4. Filtering : **Filtering** allows users to display only the required data and hide unnecessary data.

Example : In a sales visualization, selecting **2026** to display only sales data for 2026.

5. Brushing and Linking : **Brushing and Linking** allows users to select or highlight data in one visualization and see the corresponding data highlighted in another related visualization.

Example : Selecting a particular city in a sales chart can highlight the same city on a map.

GATEWAY CLASSES

GATEWAY CLASSES

6. Drill-Down

Drill-down allows users to move from a general or summarized view of data to a more detailed view.

Example : **Country → State → City → Store**

Selecting a country can show its states, and selecting a state can show its cities.

7. Interactive Selection

Interactive Selection allows users to select a particular data item, such as a bar, point, or region, to examine its details.

Example : Selecting a point in a scatter plot can display the corresponding student's study hours and marks.

VISUALIZATION SYSTEMS

- A **Visualization System** is a software tool or platform used to **represent, explore, and analyze data visually** using charts, graphs, maps, and dashboards.
- It helps users convert raw data into a form that is **easy to understand and analyze**.

Examples of Visualization Systems

- **Tableau** – Used for creating interactive charts and dashboards.
- **Microsoft Power BI** – Used for data visualization, reporting, and dashboards.
- **Qlik Sense** – Used for interactive data analysis and visualization.
- **R/RStudio** – Used for statistical analysis and creating visualizations.
- **Python Visualization Tools** – Used to create graphs and plots from data.

Features of Visualization Systems

- Represent data using charts and graphs.
- Allow users to filter and explore data.
- Help identify patterns and trends.
- Support interactive analysis.
- Provide dashboards and reports.

VISUALIZATION SYSTEMS AND APPLICATIONS

- **Business Analysis –** Helps businesses understand sales, profit, and customer trends.
- **Healthcare –** Used to analyze patient records, diseases, and medical data.
- **Finance –** Helps analyze stock prices, investments, and financial trends.
- **Education –** Used to analyze student performance, attendance, and results.
- **Scientific Research –** Helps represent experimental results and research data.
- **Marketing –** Used to understand customer behavior and market trends.
- **Big Data Analysis –** Helps understand large and complex datasets through visual representations.

ADVANTAGES OF VISUALIZATION SYSTEMS AND APPLICATIONS

- Makes complex data easy to understand.
- Helps identify patterns and trends.
- Makes comparison easier.
- Supports data-driven decisions.

INTRODUCTION TO R

R is an open-source programming language and software environment used for **statistical computing, data analysis, and visualization**.

A **Graphical User Interface (GUI)** provides a user-friendly environment to work with R without using only the command line.

```
x <- c(10, 20, 30, 40)
mean(x)
```

Main Components of R GUI

- **Console:** Used to enter and execute R commands.
- **Script Editor:** Used to write and save R programs.
- **Environment/Workspace:** Shows created variables and data.
- **Plots:** Displays graphs and visualizations.
- **Packages:** Provides additional functions and tools.

GATEWAY CLASSES

DATA IMPORT

Import means bringing data from an external file into R.

Common data sources:

- CSV files
- Text files
- Excel files
- Databases

Example

data - read.csv("student.csv")

This imports a CSV file into R.

GATEWAY CLASSES

DATA EXPORT

Export means saving R data or results into an external file.

Example:

write.csv(data, "output.csv")

GATEWAY CLASSES

ATTRIBUTES

GATEWAY CLASSES

Attributes are additional information associated with an R object.

Common attributes include:

- **Names**
- **Dimensions**
- **Class**
- **Length**

Example:

x - c(10, 20, 30)

length(x)

It gives the number of elements in x.

GATEWAY CLASSES

DATA TYPES

R supports different types of data.

1. Numeric : Stores numbers.

x - 25.5

2. Integer : Stores whole numbers.

x - 25L

3. Character : Stores text.

name - "R Programming"

4. Logical : Stores TRUE or FALSE.

x - TRUE

5. Complex : Stores complex numbers.

x - 2 + 3i

IMPORTANT DATA STRUCTURES

GATEWAY CLASSES

- **Vector:** Collection of values of the same type.
- **Matrix:** Two-dimensional data structure.
- **Array:** Multi-dimensional data structure.
- **List:** Collection of different types of objects.
- **Data Frame:** Table-like structure containing rows and columns.
- **Factor:** Used to represent categorical data.

GW
GATEWAY CLASSES

DESCRIPTIVE STATISTICS

GW
GATEWAY CLASSES

Descriptive statistics are methods used to summarize and describe data.

MAIN MEASURES

1. **Mean :** Average value of data. **Example :** mean(x)
2. **Median :** Middle value when data is arranged in order. **Example :** median(x)
3. **Mode :** The most frequently occurring value.
4. **Minimum and Maximum :** min(x), max(x)
5. **Range :** Difference between maximum and minimum value. **Example :** range(x)
6. **Variance :** Measures how much the values vary from the mean. **Example :** var(x)
7. **Standard Deviation :** Measures the spread of data around the mean. **Example :** sd(x)

EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) is the process of examining and understanding a dataset before performing detailed analysis or building models.

Main Objectives

- Understand the data.
- Find patterns and trends.
- Identify missing values.
- Detect outliers.
- Understand relationships between variables.
- Check the distribution of data.

Common EDA Methods

- Summary statistics
- Tables
- Histograms
- Box plots
- Scatter plots
- Correlation analysis

VISUALIZATION BEFORE ANALYSIS

Visualization before analysis means **using graphs and charts to understand the data before applying detailed analytical methods**.

It helps the analyst understand the dataset quickly.

Common Visualizations

- **Histogram :** Shows data distribution.
- **Box Plot :** Shows spread and outliers.
- **Scatter Plot :** Shows relationship between two variables.
- **Bar Chart :** Compares categories.
- **Line Chart :** Shows trends over time.

Example

Suppose we have students' marks. A histogram can show whether most students have **low, medium, or high marks** before further analysis.

GW
GATEWAY CLASSES

ANALYTICS FOR UNSTRUCTURED DATA

GW
GATEWAY CLASSES

- Unstructured data is data that does not have a fixed format or tabular structure.
- It is the process of extracting useful information, patterns and insights from unstructured data using data analysis techniques.

Examples : Text, emails, social media posts, images, audio and videos.

R-Based Workflow for Text Data

- **Data Collection/Import –** Collect or import text data into R.
- **Preprocessing –** Clean the text by removing unnecessary symbols, punctuation, stop words, etc.
- **Analysis –** Identify useful information such as word frequency, keywords, topics or sentiment.
- **Visualization –** Represent results using graphs, frequency plots or word clouds.
- **Result/Insights –** Interpret the results and obtain useful information.

GATEWAY CLASSES

GATEWAY CLASSES

Applications

- **Sentiment Analysis –** Analyze opinions in reviews/social media.
- **Text Classification –** Classify documents or messages.
- **Keyword Analysis –** Find important words.
- **Topic Analysis –** Identify major topics in documents.

Gateway Classes

@GATEWAY CLASSES

Empowering Learners, Transforming Futures

THANK YOU

GET IT ON Google Play

Download on the App Store

Helpline No- 7819 0058 53 , 9259833200