Definition of Big Data

- Big Data refers to **very large, complex, and rapidly generated datasets**
- Cannot be **stored, processed, or analyzed** using traditional database systems
- Includes:
  - **Structured data**
  - **Semi-structured data**
  - **Unstructured data**
- Requires **distributed storage and parallel processing**
- Characterized by **5 V's**:
  - Volume
  - Velocity
  - Variety
  - Veracity
  - Value

Evolution of Big Data (Main 4 Eras)

1. Traditional Data Era (Before 1990s)

- Data was:
  - Small in size
  - Highly structured
  - Centrally stored
- Technologies used:
  - File systems
  - Relational Database Management Systems (RDBMS)

    - SQL
- Examples:
    - Banking transaction records
    - Payroll systems
    - Student information systems
- Limitations:
    - No support for unstructured data
    - Limited scalability
    - High cost of system expansion

---

2. Data Warehousing & Business Intelligence Era (1990s – Early 2000s)

- Growth of enterprise-level data
- Focus on **historical data analysis**
- Technologies used:
    - Data warehouses
    - ETL (Extract, Transform, Load) tools
    - OLAP systems
- Features:
    - Centralized storage
    - Batch processing
    - Structured reporting
- Examples:
    - Sales analysis
    - Financial reporting
    - Customer profiling

- Limitations:
  - Expensive infrastructure
  - Poor handling of unstructured data
  - No real-time analytics

---

3. Web and Social Media Data Era (Mid-2000s)

- Rapid increase in internet usage
- Data became:
  - Huge in volume
  - Mostly unstructured
  - Continuously generated
- Data sources:
  - Social media platforms
  - Websites
  - Multimedia content
- Examples:
  - Facebook posts and likes
  - YouTube videos
  - Web server logs
- Challenges:
  - Traditional databases failed to scale
  - Storage and processing inefficiency

---

4. Big Data Era (Hadoop & Real-Time Analytics) (Late 2000s – Present)

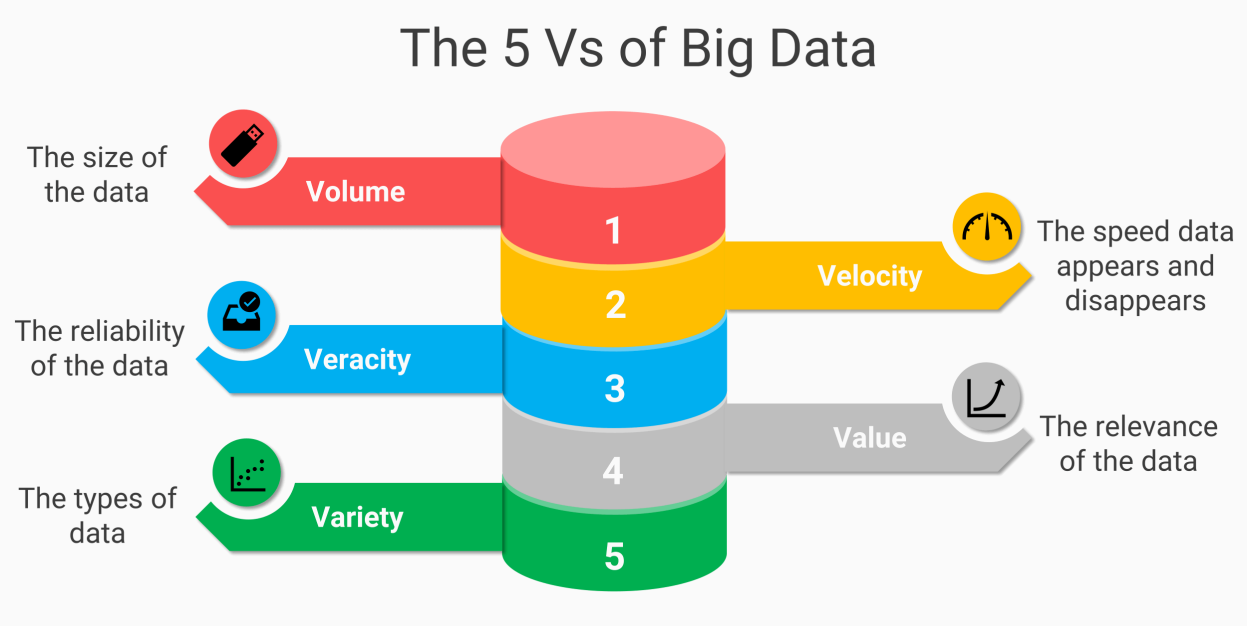
- Introduction of **distributed computing**
- Technologies used:

    - Hadoop (HDFS, MapReduce)
    - Apache Spark
    - NoSQL databases
    - Cloud platforms
- Features:
    - Distributed storage
    - Parallel processing
    - Fault tolerance
    - Real-time analytics
- Examples:
    - Recommendation systems (Netflix, Amazon)
    - Fraud detection in banking
    - IoT sensor data analysis
    - Search engine indexing

Conclusion

- Big Data evolved from:
    - Small, structured data systems
    - To large, distributed, real-time data environments
- Modern Big Data technologies enable:
    - Faster decision-making
    - Advanced analytics
    - AI and machine learning applications

Characteristics of Big Data – 5 V's (VTU Style, Detailed)



1. Volume

- Refers to the **huge amount of data** generated and stored
- Data size ranges from:
  - Terabytes (TB)
  - Petabytes (PB)
  - Exabytes (EB) and beyond
- Generated from:
  - Social media platforms
  - Sensors and IoT devices
  - Transaction systems
  - Multimedia (images, videos)
- Traditional databases cannot handle such massive data volumes

- Requires:
  - Distributed storage systems
  - Scalable architectures

Example:

- Facebook generates terabytes of user data daily
- E-commerce websites store millions of transaction records

---

2. Velocity

- Refers to the **speed at which data is generated, transmitted, and processed**
- Data is produced:
  - Continuously
  - In real time or near real time
- High velocity data needs **fast processing and quick decision-making**
- Batch processing is often insufficient

Sources of high-velocity data:

- Stock market transactions
- Online payments
- Live social media feeds
- Sensor data from IoT devices

Example:

- Real-time fraud detection in banking
- Live traffic updates in navigation apps

---

3. Variety

- Refers to **different types and formats of data**
- Big Data includes:

    - **Structured data** (tables, rows, columns)
    - **Semi-structured data** (XML, JSON, logs)
    - **Unstructured data** (text, images, videos, audio)
- Traditional systems are designed mainly for structured data
- Big Data systems can handle all formats

Example:

- Emails (text)
- Social media posts (text + images)
- CCTV videos
- Server log files

---

4. Veracity

- Refers to the **quality, accuracy, and reliability of data**
- Big Data often contains:
    - Noise
    - Duplicates
    - Incomplete or inconsistent data
- Poor data quality can lead to:
    - Incorrect analysis
    - Wrong business decisions
- Data cleaning and validation are essential

Challenges:

- Fake social media data
- Sensor errors
- Human-generated data inconsistencies

Example:

- Fake reviews on e-commerce platforms
- Incorrect readings from faulty sensors

---

5. Value

- Refers to the **usefulness and meaningful insights** extracted from data
- Large volumes of data are useless without proper analysis
- The main goal of Big Data analytics is to:
  - Discover patterns
  - Predict trends
  - Support decision-making
- Value depends on effective analytics and interpretation

Example:

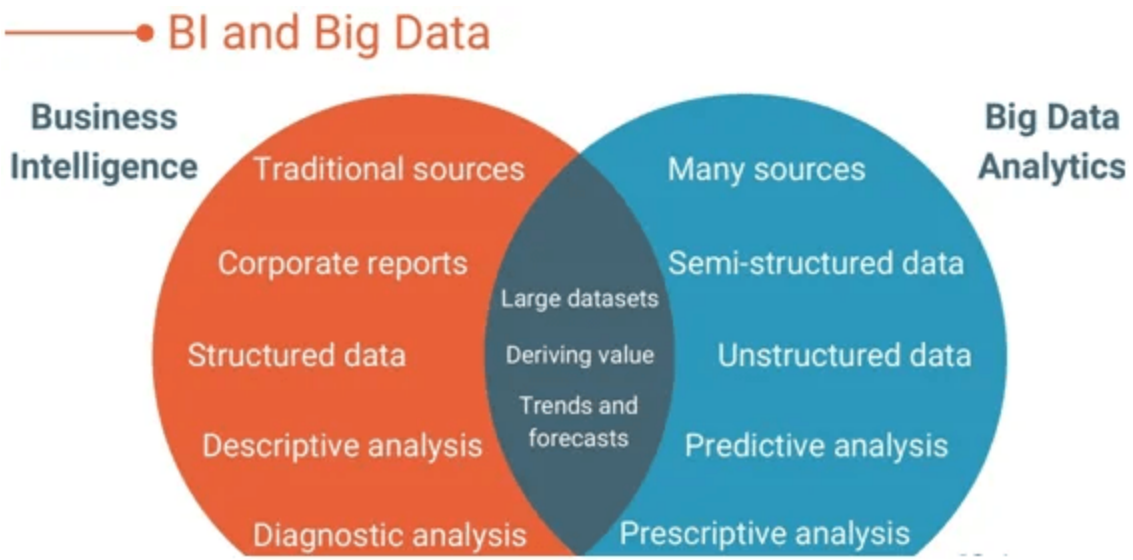
- Customer behavior analysis for targeted marketing
- Predictive maintenance in industries
- Personalized recommendations (Netflix, Amazon)

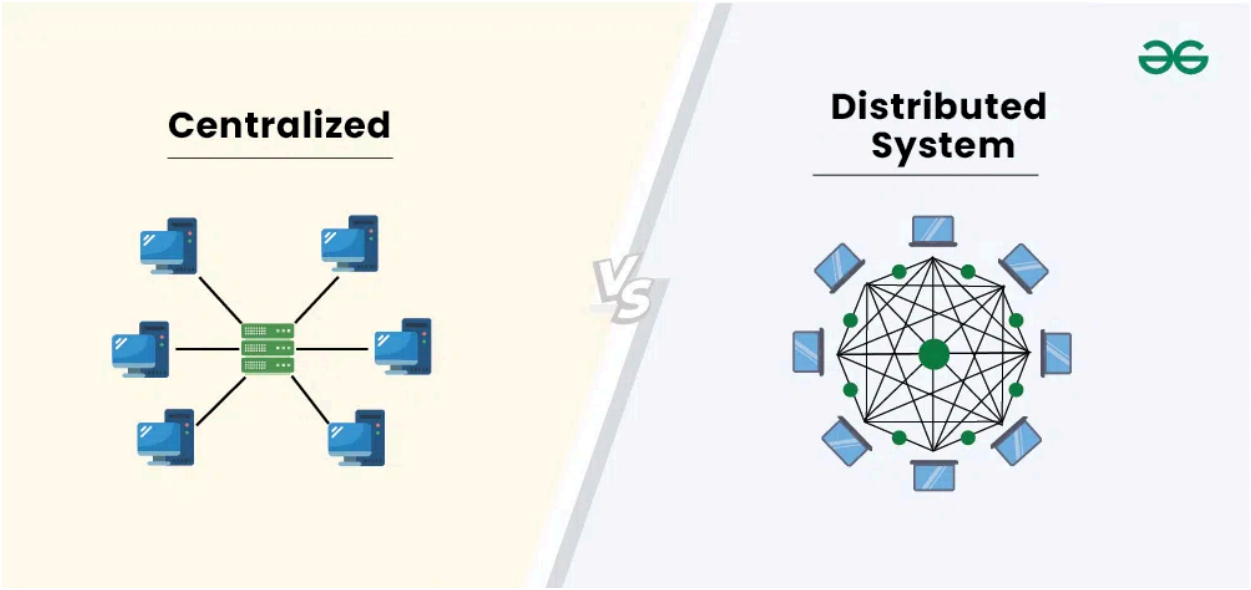
Conclusion

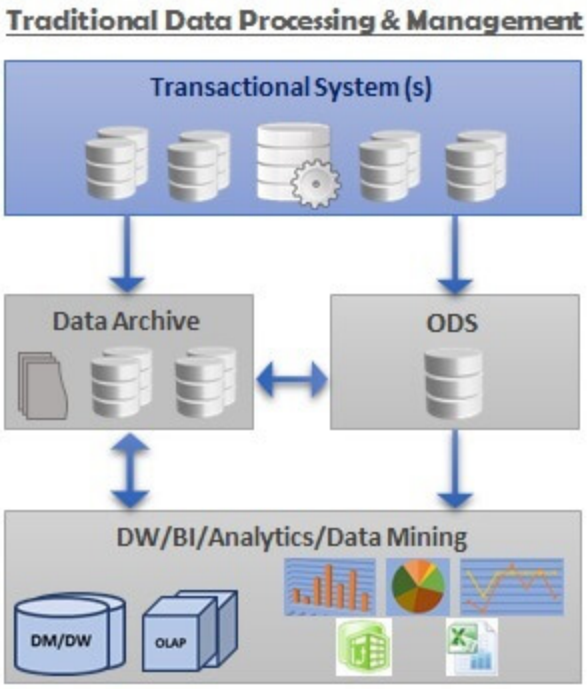
- The **5 V's** describe the fundamental characteristics of Big Data
- They explain why traditional systems are insufficient
- Understanding these characteristics helps in:
  - Designing Big Data systems
  - Choosing appropriate tools and technologies
  - Extracting maximum business and analytical value

Comparison of Traditional Data Processing Systems and Big Data Systems

Traditional Data Processing Systems and Big Data Systems differ significantly in terms of data size, data types, architecture, and processing methods. Traditional systems were designed to handle structured, limited-volume data, whereas Big Data systems are built to manage massive, fast, and diverse datasets generated in modern digital environments.







1. Traditional Data Processing Systems (Detailed Explanation)

Traditional data processing systems were developed to manage data generated by early business and enterprise applications.

- Designed to handle **small to moderate volumes of structured data**
- Data is stored in **relational databases (RDBMS)** using tables, rows, and columns
- Uses **centralized architecture** (single server or limited servers)
- Processing is mainly **batch-oriented**
- Uses **SQL** for querying and manipulation
- Scaling is done by **vertical scaling** (upgrading hardware)
- Limited support for unstructured or semi-structured data
- High cost when data volume increases

Examples:

- Banking transaction systems
- Payroll and accounting systems
- Student information systems

---

2. Big Data Systems (Detailed Explanation)

Big Data systems were introduced to overcome the limitations of traditional systems and handle modern data challenges.

- Designed to handle **very large volumes of data (TBs–EBs)**
- Supports **structured, semi-structured, and unstructured data**
- Uses **distributed architecture** (cluster of commodity machines)
- Supports **batch processing and real-time/stream processing**
- Uses frameworks like Hadoop and Spark
- Scaling is done by **horizontal scaling** (adding more nodes)
- Built-in **fault tolerance and high availability**
- Cost-effective for large-scale data processing

Examples:

- Social media data analysis
- Recommendation systems (Netflix, Amazon)
- IoT sensor data processing
- Real-time fraud detection

---

3. Column-wise Comparison

Aspect	Traditional Data Processing Systems	Big Data Systems
Data Volume	Small to moderate	Very large (TBs–EBs)
Data Type	Structured only	Structured, semi-structured, unstructured

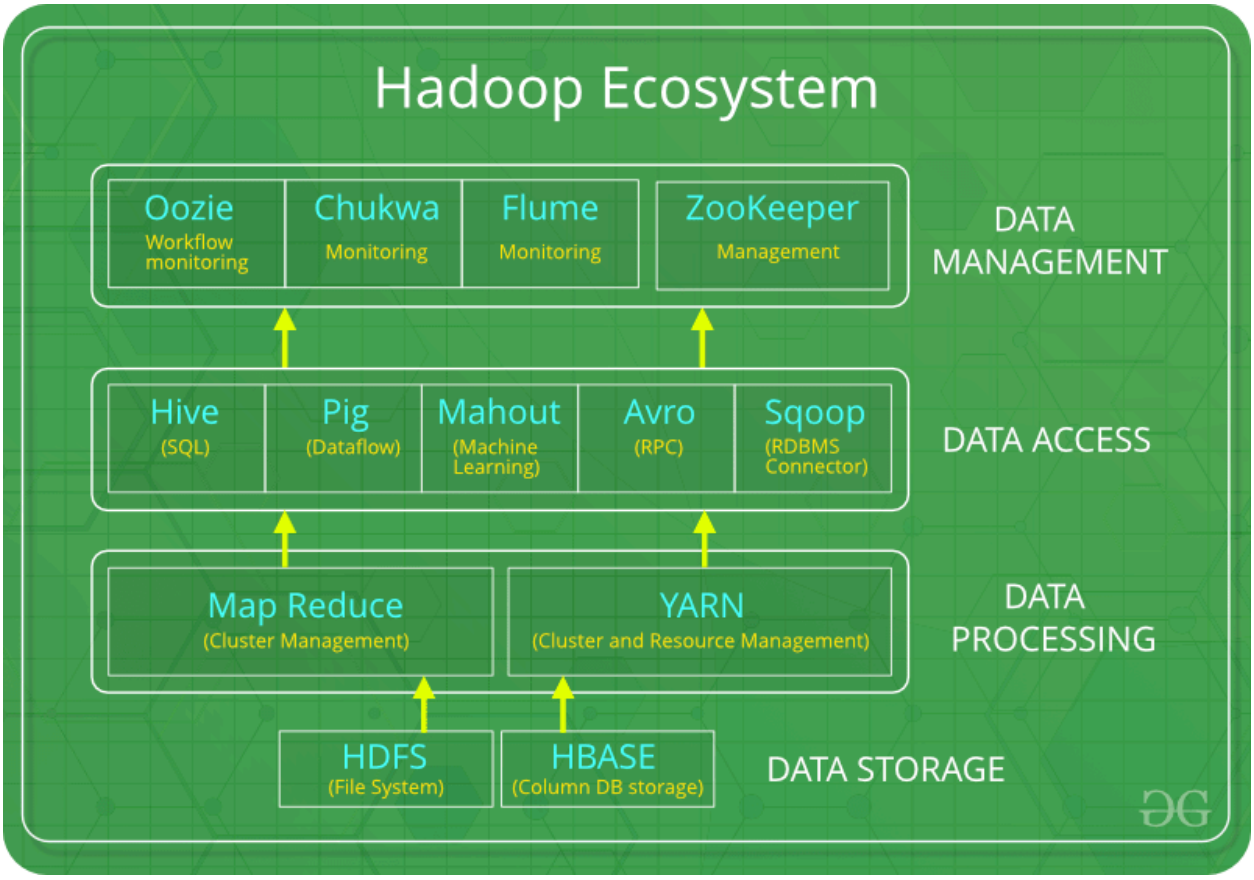
Aspect	Traditional Data Processing Systems	Big Data Systems
Architecture	Centralized	Distributed
Storage	Relational databases	Distributed file systems and NoSQL
Processing	Batch processing	Batch and real-time processing
Scalability	Vertical scaling	Horizontal scaling
Fault Tolerance	Limited	High (replication, recovery)
Cost	Expensive at large scale	Cost-effective using commodity hardware
Flexibility	Low	High
Examples	Payroll, banking systems	Social media, IoT, analytics platforms

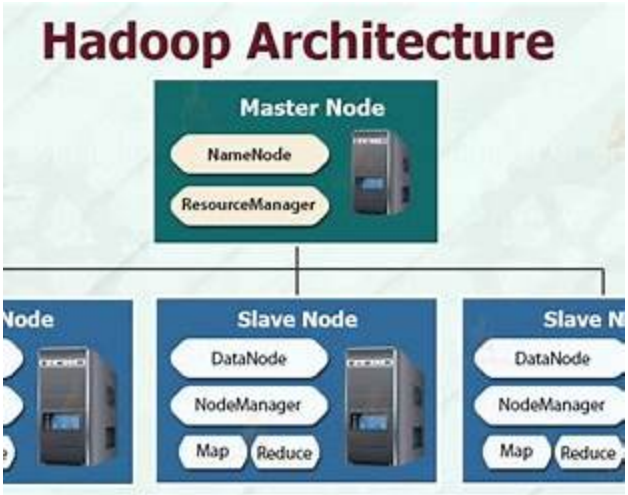
Conclusion

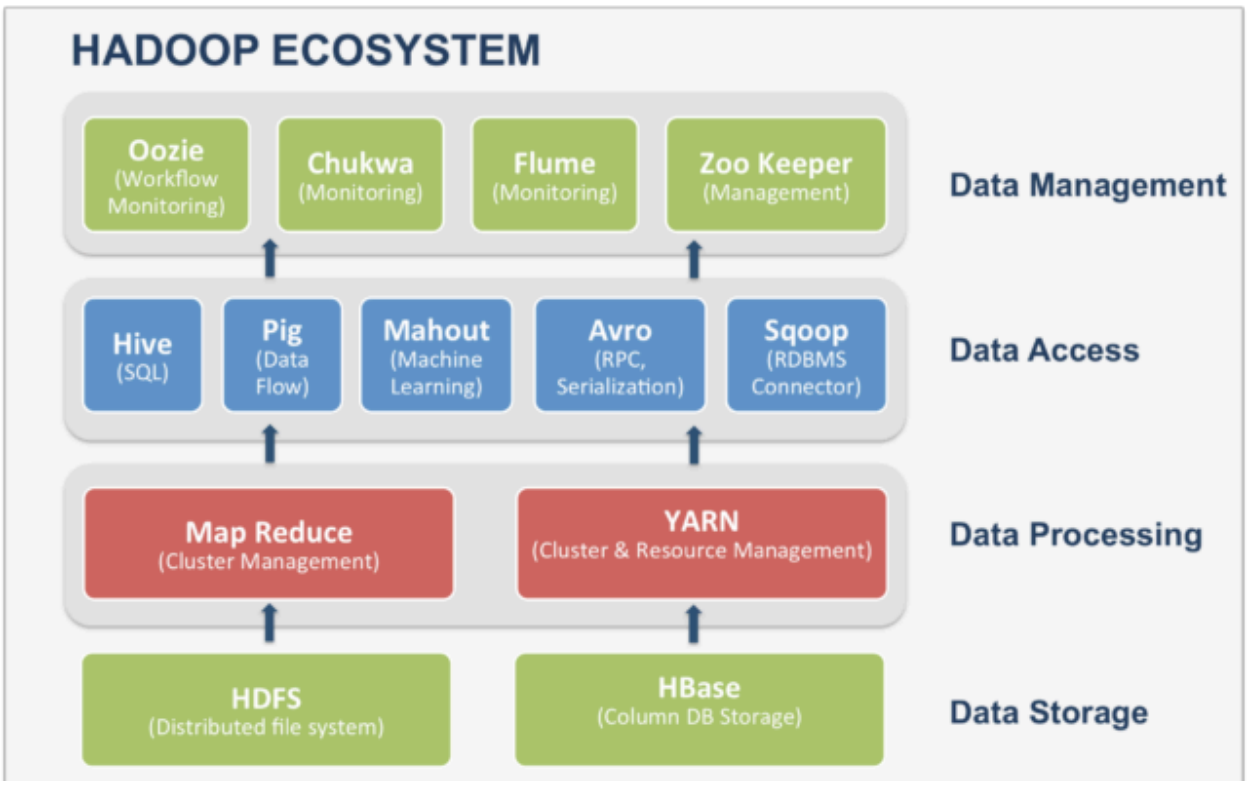
- Traditional systems are suitable for **small, structured, and stable data**
- Big Data systems are designed for **large-scale, fast, and diverse data**
- Modern applications require Big Data systems for **scalability, flexibility, and real-time insights**

Hadoop Ecosystem – Explanation with Architecture Diagram

The **Hadoop ecosystem** is a collection of **open-source tools and frameworks** designed to store, process, manage, and analyze **large-scale data** efficiently using distributed computing. Hadoop provides a scalable, fault-tolerant, and cost-effective solution for handling Big Data across clusters of commodity hardware.


Hadoop Ecosystem
Oozie
Workflow monitoring
Chukwa
Monitoring
Flume
Monitoring
ZooKeeper
Management
DATA MANAGEMENT
Hive
(SQL)
Pig
(Dataflow)
Mahout
(Machine Learning)
Avro
(RPC)
Sqoop
(RDBMS Connector)
DATA ACCESS
Map Reduce
(Cluster Management)
YARN
(Cluster and Resource Management)
DATA PROCESSING
HDFS
(File System)
HBASE
(Column DB storage)
DATA STORAGE


Hadoop Architecture
Master Node
NameNode
ResourceManager
Slave Node
DataNode
NodeManager
Map
Reduce
Slave N
DataNode
NodeManager
Map
Reduce



Core Components of Hadoop Ecosystem

1. Hadoop Distributed File System (HDFS)

HDFS is the **storage layer** of Hadoop.

- Stores very large files across multiple machines
- Uses **block-based storage**
- Provides **data replication** for fault tolerance
- Designed for high-throughput access

Key components:

- NameNode – Manages metadata
- DataNode – Stores actual data blocks
- Secondary NameNode – Assists NameNode

2. MapReduce

MapReduce is the **processing layer** of Hadoop.

- Programming model for distributed data processing
- Divides tasks into:
  - Map phase – processes input data
  - Reduce phase – aggregates results
- Enables parallel processing across nodes
- Suitable for batch processing

---

3. YARN (Yet Another Resource Negotiator)

YARN is the **resource management layer**.

- Manages cluster resources (CPU, memory)
- Schedules jobs efficiently
- Allows multiple processing frameworks to run on Hadoop
- Improves scalability and performance

Components:

- ResourceManager
- NodeManager
- ApplicationMaster

---

4. Hadoop Common

- Collection of **shared utilities and libraries**
- Supports other Hadoop modules
- Includes configuration files and Java libraries

Supporting Tools in Hadoop Ecosystem

5. Hive

- Data warehousing tool
- Uses **SQL-like language (HiveQL)**
- Suitable for analytical queries

6. Pig

- High-level scripting language (Pig Latin)
- Used for data transformation and ETL tasks

7. HBase

- NoSQL column-oriented database
- Provides real-time read/write access
- Runs on top of HDFS

8. Sqoop

- Transfers data between RDBMS and HDFS
- Supports import and export operations

9. Flume

- Collects and transfers streaming data
- Used for log data ingestion

10. Oozie

- Workflow scheduler
- Manages Hadoop job dependencies

Hadoop Ecosystem Architecture (Explanation)

- **Data Sources**: RDBMS, logs, sensors, social media
- **Ingestion Layer**: Flume, Sqoop

- **Storage Layer**: HDFS, HBase
- **Processing Layer**: MapReduce, Spark
- **Resource Management**: YARN
- **Analytics Layer**: Hive, Pig
- **Workflow & Management**: Oozie, ZooKeeper

Advantages of Hadoop Ecosystem

- Scalable and distributed
- Fault-tolerant
- Cost-effective
- Supports diverse data formats
- Suitable for Big Data analytics