UNIT–3 : DATA IN THE CLOUD & MAP REDUCE

Subject: Cloud Computing

RGPV Exam Notes (One-Night Preparation Notes)

UNIT OVERVIEW

This unit is one of the **MOST IMPORTANT UNITS** in Cloud Computing.

Questions from this unit come regularly in RGPV exams because it contains:

✅ Big Data
✅ Google File Systems
✅ Hadoop
✅ HDFS
✅ Map Reduce
✅ Parallel Processing
✅ Dynamo
✅ Big Table

These topics are directly related to modern companies like:

- Google
- Amazon
- Facebook
- Netflix

- YouTube

HOW TO STUDY THIS UNIT FAST

Remember this sequence:

```
Huge Data Problem
        ↓
Need Distributed Storage
        ↓
GFS / HDFS
        ↓
Need Database for Big Data
        ↓
BigTable / HBase / Dynamo
        ↓
Need Fast Processing
        ↓
Map Reduce
```

This flow itself can help you write introductions in exams.

---

1. RELATIONAL DATABASES IN CLOUD

Definition

A Relational Database is a database that stores data in the form of tables containing rows and columns.

Examples:

- MySQL

- Oracle
- SQL Server
- PostgreSQL

Easy Introduction

In traditional systems, data is stored in tables.

Cloud platforms also use databases to store huge amounts of user data.

Example:

- Student records
- Banking data
- E-commerce orders

Cloud databases allow:

- online access
- scalability
- backup
- remote management

Why Important

Cloud applications need:

- fast data access
- large storage
- reliability

Relational databases provide:

- structured storage
- easy querying using SQL

Detailed Explanation

A relational database organizes data into:

Table	Meaning
Rows	Records
Columns	Attributes

Example:

Roll No	Name	Branch
101	Rahul	CSE
102	Aman	IT

Data can be related using keys.

Example:

- Primary Key
- Foreign Key

Features of Cloud Relational Databases

- Data stored on cloud servers
- Accessible from anywhere
- Automatic backup
- High availability
- Scalable storage

Examples

Service	Company
Amazon RDS	Amazon
Cloud SQL	Google
Azure SQL	Microsoft

Advantages

- Easy data management
- Structured storage
- SQL support
- High security
- Backup and recovery

Disadvantages

- Difficult for unstructured data
- Expensive for huge big data
- Scaling limitations

Applications

- Banking systems
- College ERP
- Hospital management
- E-commerce websites

Important Keywords

- Structured Data

- Tables
- Rows and Columns
- SQL
- Primary Key
- Scalability

Conclusion

Relational databases are used for structured data storage in cloud environments and provide reliable and secure database management.

---

2. CLOUD FILE SYSTEMS

Definition

A Cloud File System stores files across multiple distributed servers connected through a network.

Easy Introduction

Normal computers store files in one hard disk.

But companies like Google store data in thousands of systems.

So they need special file systems.

Examples:

- GFS
- HDFS

Why Needed

Huge companies handle:

- videos
- photos
- emails
- logs

Single storage system cannot manage such huge data.

Therefore distributed file systems are used.

Types of Cloud File Systems

1. GFS (Google File System)
2. HDFS (Hadoop Distributed File System)

---

3. GOOGLE FILE SYSTEM (GFS)

Definition

GFS is a distributed file system developed by Google for storing and processing huge amounts of data across multiple machines.

Easy Explanation

Google handles:

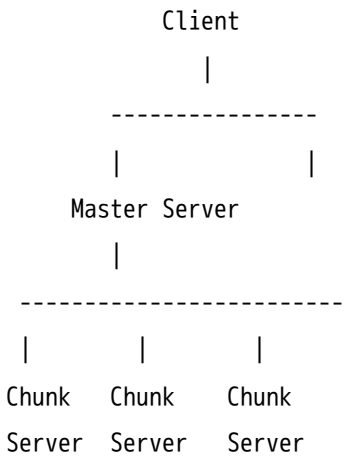
- YouTube videos

- Gmail data
- Search engine data

One computer cannot store all this.

So Google divides data into chunks and stores them on many servers.

Architecture of GFS



Components

Component	Work
Master Server	Controls metadata
Chunk Server	Stores actual data
Client	Requests data

Working of GFS

Step 1:

File divided into chunks.

Step 2:

Chunks stored on chunk servers.

Step 3:

Master stores metadata.

Step 4:

Client accesses data through master.

Features of GFS

- Large file support
- Fault tolerance
- High scalability
- Replication support

Advantages

- Fast processing
- Reliable
- Handles big data efficiently

Disadvantages

- Complex architecture
- Single master failure issue

Applications

- Google Search
- YouTube
- Gmail

Important Keywords

- Chunk Server
- Metadata
- Replication
- Distributed Storage

---

4. HDFS (Hadoop Distributed File System)

Definition

HDFS is a distributed file system used in Hadoop to store very large datasets across multiple computers.

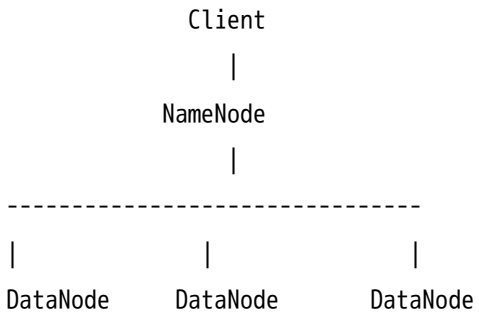
Easy Introduction

HDFS is inspired by GFS.

It stores data in blocks across many systems.

Used for Big Data analytics.

HDFS Architecture



Components

Component	Function
NameNode	Stores metadata
DataNode	Stores actual blocks
Client	Access data

Working of HDFS

Step 1:

File divided into blocks.

Step 2:

Blocks stored in DataNodes.

Step 3:

NameNode manages block information.

Step 4:

Replication used for fault tolerance.

Features

- Distributed storage
- Fault tolerance
- High throughput
- Replication

Advantages

- Cheap storage
- Reliable
- Scalable

Disadvantages

- NameNode failure problem
- Not suitable for small files

Applications

- Hadoop ecosystem
- Big data analytics
- Data mining

Important Keywords

- NameNode
- DataNode

- Replication
- Fault Tolerance

---

5. DIFFERENCE BETWEEN GFS & HDFS

Feature	GFS	HDFS
Developed By	Google	Apache Hadoop
Master Node	Master Server	NameNode
Storage Unit	Chunks	Blocks
Open Source	No	Yes
Usage	Google Internal	Hadoop Systems

Memory Trick

```
Google → GFS
Hadoop → HDFS
Master → NameNode
Chunk → Block
```

---

6. BIG TABLE

Definition

BigTable is a distributed storage system developed by Google for managing structured data.

Easy Introduction

Google stores billions of web pages.

Traditional databases cannot handle such huge data.

So Google created BigTable.

Features

- Distributed database
- Scalable
- High performance
- Column-oriented storage

Structure

```
Row Key → Column → Timestamp → Value
```

Applications

- Google Earth
- Google Search
- Gmail

Advantages

- Fast access
- Handles huge data

- Highly scalable

Disadvantages

- Complex design
- Not suitable for small systems

Important Keywords

- Structured Data
- Column Family
- Distributed Database

---

7. HBASE

Definition

HBase is an open-source distributed database built on top of HDFS.

Easy Explanation

HBase is inspired by BigTable.

Used for storing massive datasets in Hadoop systems.

Features

- Column-oriented
- Real-time access
- Distributed database

Architecture

```
Client
   |
HMaster
   |
Region Servers
   |
HDFS
```

Advantages

- Real-time processing
- Scalable
- Handles big data

Disadvantages

- Complex management
- High memory usage

Applications

- Social media analytics

- Banking analytics
- Log analysis

---

8. DYNAMO

Definition

Dynamo is a highly available distributed key-value database developed by Amazon.

Easy Introduction

Amazon needs databases that never stop.

Even if one server fails, shopping must continue.

So Amazon created Dynamo.

Features

- Distributed storage
- High availability
- Fault tolerance
- Key-value storage

Working

```
Key → Value
```

Example:

```
101 → Laptop
102 → Mobile
```

Advantages

- Highly reliable
- Fast response
- No single point failure

Disadvantages

- Complex synchronization
- Eventual consistency problem

Applications

- Amazon shopping systems
- Online transaction systems

---

9. MAP REDUCE

Definition

Map Reduce is a programming model used for processing large datasets in parallel across distributed systems.

Easy Introduction

Suppose Google wants to count words from billions of pages.

One computer will take too much time.

So work is divided among many computers.

This process is called Map Reduce.

Basic Idea

```
Large Task
     ↓
Divide into Small Tasks
     ↓
Process in Parallel
     ↓
Combine Result
```

Architecture

```
Input Data
     |
   MAP
     |
Intermediate Data
```

```
    |
REDUCE
    |
Final Output
```

MAP Phase

- Divides task
- Processes small parts

Example:

Count words.

Input:

```
Cloud Computing is Easy
```

Map Output:

```
Cloud → 1
Computing → 1
is → 1
Easy → 1
```

REDUCE Phase

Combines same keys.

Final Output:

```
Cloud → 1
Computing → 1
Easy → 1
```

Detailed Working

Step 1:

Input data divided.

Step 2:

Map function processes data.

Step 3:

Intermediate output generated.

Step 4:

Reduce combines results.

Step 5:

Final output produced.

Advantages

- Parallel processing
- Faster execution
- Handles huge data
- Fault tolerant

Disadvantages

- Complex debugging
- Not suitable for small tasks

Applications

- Search engines
- Log analysis
- Data mining
- Recommendation systems

Important Keywords

- Parallel Processing
- Distributed Computing
- Mapper
- Reducer
- Big Data

---

10. PARALLEL COMPUTING

Definition

Parallel computing means performing multiple computations simultaneously using multiple processors.

Easy Introduction

Instead of one person doing all work,
many people work together.

Same concept in computers.

Example

1000 files processed by:

- 1 processor → slow
- 10 processors → fast

Advantages

- Faster execution
- Better performance
- Time saving

Applications

- AI
- Big data
- Scientific simulations

---

11. PARALLEL EFFICIENCY OF MAP REDUCE

Definition

Parallel efficiency measures how efficiently multiple systems work together during Map Reduce processing.

Formula

Parallel Efficiency=SpeedupNumber of Processors\text{Parallel Efficiency} = \frac{\text{Speedup}}{\text{Number of Processors}}Parallel Efficiency=Number of ProcessorsSpeedup

---

Easy Explanation

If 10 processors are used and work becomes 8 times faster:

Efficiency:

Efficiency=810=0.8\text{Efficiency} = \frac{8}{10} = 0.8Efficiency=108=0.8

Meaning:
80% efficiency.

---

12. RELATIONAL OPERATIONS IN MAP REDUCE

Operations performed:

- Selection
- Projection
- Join
- Group By

Used for big database processing.

---

13. ENTERPRISE BATCH PROCESSING

Definition

Batch processing means processing large amounts of data together at scheduled times.

Examples

- Salary generation
- Electricity bill generation
- Exam result processing

Advantages

- Saves time
- Handles bulk data
- Efficient processing

---

14. EXAMPLES / APPLICATIONS OF MAP REDUCE

Application	Usage
Google Search	Indexing
Facebook	Analytics
Amazon	Recommendation
Banking	Fraud detection
YouTube	Video analysis

UNIT–3 IMPORTANT DIAGRAMS

GFS Diagram

```
Client → Master → Chunk Servers
```

HDFS Diagram

```
Client → NameNode → DataNodes
```

Map Reduce Diagram

```
Input → Map → Shuffle → Reduce → Output
```

🔥 MOST IMPORTANT TOPICS

⭐ HDFS
⭐ GFS
⭐ Map Reduce
⭐ BigTable
⭐ Dynamo

⭐ Difference between GFS and HDFS

⭐ Parallel Computing

⭐ MOST IMPORTANT 7-MARK QUESTIONS

1. Explain GFS architecture.
2. Explain HDFS working.
3. What is Map Reduce? Explain with example.
4. Explain BigTable.
5. Explain Dynamo database.
6. Difference between GFS and HDFS.
7. Explain parallel computing.

⭐ MOST IMPORTANT 14-MARK QUESTIONS

1. Explain HDFS architecture and working in detail.
2. Explain Map Reduce model with diagram and example.
3. Compare GFS and HDFS in detail.
4. Explain BigTable, HBase and Dynamo.
5. Explain applications of Map Reduce.

🎯 PYQ-BASED EXPECTED QUESTIONS

⭐ Very High Probability

✔ HDFS Architecture

✔ Map Reduce Working

✔ GFS vs HDFS

✔ BigTable

✔ Dynamo

⭐ High Probability

✔ Parallel Computing

✔ HBase

✔ Enterprise Batch Processing

⭐ Medium Probability

✔ Relational Operations

✔ Parallel Efficiency

⚡ ONE-NIGHT REVISION NOTES

```
GFS → Google storage system

HDFS → Hadoop storage system

NameNode → Metadata

DataNode → Actual data

Map → Divide work

Reduce → Combine results

BigTable → Google database
```

```
HBase → Hadoop database

Dynamo → Amazon key-value database

Parallel Computing → Many processors work together
```

📌 SMART 2-HOUR REVISION STRATEGY

First 30 Minutes

Read:

- GFS
- HDFS

Next 40 Minutes

Read:

- Map Reduce
- Parallel Computing

Next 20 Minutes

Read:

- BigTable
- HBase
- Dynamo

Last 30 Minutes

Revise:

- diagrams
- differences
- keywords

📌 5-HOUR PREPARATION STRATEGY

Time	Topic
1 Hour	GFS + HDFS
1.5 Hour	Map Reduce
45 Min	BigTable + HBase
30 Min	Dynamo
30 Min	Parallel Computing
45 Min	PYQs + Revision

🌙 ONE-NIGHT PREPARATION PLAN

If very less time:

Priority Order:

1. HDFS
2. Map Reduce
3. GFS
4. BigTable
5. Dynamo

These alone can help score good marks.

🚀 TOPPER ANSWER WRITING TIPS

1. Always Start with Definition

Examiner loves proper definitions.

---

2. Draw Diagram

Even simple text diagrams increase marks.

---

3. Underline Keywords

Underline:

- Fault Tolerance
- Scalability
- Distributed System
- Replication
- Parallel Processing

---

4. Fill Answer Smartly

Structure:

```
Definition
Introduction
Diagram
Working
Advantages
Disadvantages
Applications
Conclusion
```

---

5. Write Real-Life Examples

Examples make answers look practical.

FINAL REVISION MANTRA 🚀

```
Storage → GFS/HDFS
Database → BigTable/HBase/Dynamo
Processing → Map Reduce
Speed → Parallel Computing
```

If you remember this flow,

you can write entire Unit–3 easily in exam.