Cloud Computing Unit -3 Important Questions With Easy Explaination

One-Night RGPV Exam Preparation

Topics: Map Reduce, GFS vs HDFS, BigTable, Dynamo, Parallel Computing, HBase, Batch Processing, Relational Operations, Parallel Efficiency

---

1. Map Reduce Working

Introduction

Map Reduce is used to process very large data in less time.
It divides a big task into small tasks and runs them on many computers together.

Definition

Map Reduce is a programming model used for processing large datasets in parallel on distributed computers.

Why It Is Needed

Normal computer cannot process huge data like Google search pages, YouTube logs, banking records, or social media data.
Map Reduce solves this by dividing work among many machines.

Easy Explanation

Think of checking 10,000 exam copies.
If one teacher checks all copies, it takes many days.

But if 100 teachers check copies together, work finishes fast.

This is Map Reduce.

Step-by-Step Working

Step 1: Input Data

Large data is given as input.

Step 2: Splitting

Data is divided into small parts.

Step 3: Map Phase

Each small part is processed separately.

Step 4: Shuffle and Sort

Similar outputs are grouped together.

Step 5: Reduce Phase

Grouped data is combined to produce final result.

Step 6: Output

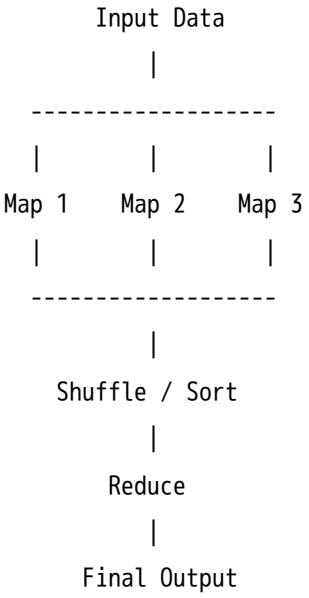
Final answer is stored.

Flow of Process

```
Large Input Data
      ↓
Split into Small Blocks
      ↓
Map Function
```

```
      ↓
Intermediate Key-Value Pairs
      ↓
Shuffle and Sort
      ↓
Reduce Function
      ↓
Final Output
```

Diagram



Real-Life Analogy

Counting votes in election:

Different teams count votes from different booths. Then results are combined to get final total.

Example

Input:

```
Cloud is easy
Cloud is powerful
```

Map Output:

```
Cloud → 1
is → 1
easy → 1
Cloud → 1
is → 1
powerful → 1
```

Reduce Output:

```
Cloud → 2
is → 2
easy → 1
powerful → 1
```

Advantages

- Processes huge data fast
- Supports parallel processing
- Fault tolerant
- Scalable
- Useful for Big Data

Disadvantages

- Not suitable for small data
- Debugging is difficult
- Real-time processing is weak
- Requires distributed setup

Applications

- Google search indexing
- Log analysis
- Data mining
- Recommendation systems
- Banking fraud detection

Important Keywords

Mapper, Reducer, Key-Value Pair, Shuffle, Sort, Parallel Processing, Distributed Computing, Big Data

Conclusion

Map Reduce is a powerful model for processing huge data by dividing work into map and reduce phases. It is very important for Big Data and distributed cloud systems.

---

2. GFS vs HDFS

Introduction

GFS and HDFS are distributed file systems.
They store huge files across multiple machines.

Definition

GFS: Google File System is a distributed file system developed by Google.

HDFS: Hadoop Distributed File System is an open-source distributed file system used in Hadoop.

Why It Is Needed

Companies store huge data. One machine cannot store and process it.

So data is divided and stored on many machines.

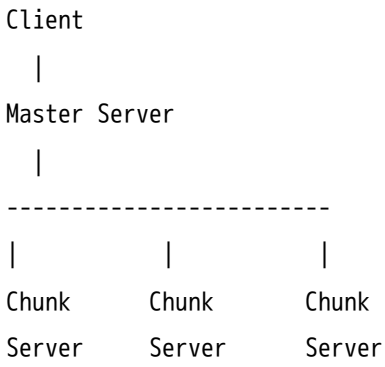
Easy Explanation

Imagine one notebook cannot contain all notes.

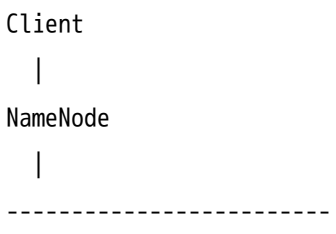
So you divide notes into many notebooks and keep backup copies.

This is how GFS and HDFS work.

GFS Architecture



HDFS Architecture



```
|           |           |
DataNode    DataNode    DataNode
```

Step-by-Step Working

GFS

1. File is divided into chunks.
2. Chunks are stored on chunk servers.
3. Master stores metadata.
4. Client asks master for location.
5. Client reads data from chunk server.

HDFS

1. File is divided into blocks.
2. Blocks are stored on DataNodes.
3. NameNode stores metadata.
4. Data is replicated for safety.
5. Client reads data from DataNodes.

Proper Comparison Table

Feature	GFS	HDFS
Full Form	Google File System	Hadoop Distributed File System
Developed By	Google	Apache Hadoop
Main Node	Master Server	NameNode
Storage Node	Chunk Server	DataNode
Data Unit	Chunk	Block
Open Source	No	Yes
Used In	Google internal systems	Hadoop Big Data systems
Fault Tolerance	Yes	Yes

Feature	GFS	HDFS
Replication	Yes	Yes
Best For	Google-scale data	Big Data analytics

Which Is Better and Why?

For companies like Google, **GFS is better** because it is specially designed for Google's internal systems.
For students, researchers, and open-source Big Data projects, **HDFS is better** because it is free, open-source, and used widely with Hadoop.

Advantages

- Stores huge data
- Fault tolerant
- Scalable
- Supports replication
- Useful for Big Data

Disadvantages

- Complex design
- Not suitable for small files
- Master/NameNode failure can create issues

Applications

- Search engines
- Big Data analytics
- Cloud storage
- Log storage
- Data mining

Important Keywords

Distributed File System, Metadata, Replication, Fault Tolerance, Chunk Server, NameNode, DataNode

Conclusion

GFS and HDFS are distributed file systems used to store huge data. GFS is Google’s system, while HDFS is Hadoop’s open-source file system.

Memory Trick

```
GFS = Google + Chunk Server
HDFS = Hadoop + DataNode
```

---

3. BigTable

Introduction

BigTable is developed by Google for storing huge structured data.
It is not a normal SQL database.

Definition

BigTable is a distributed, scalable, column-oriented storage system developed by Google for managing large structured data.

Why It Is Needed

Normal databases cannot easily manage billions of rows and huge web data.

BigTable stores large data efficiently.

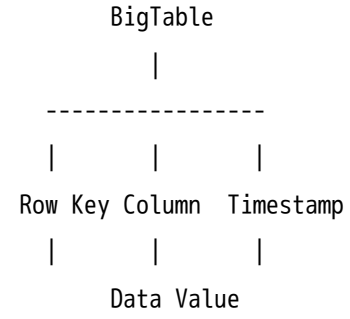
Easy Explanation

BigTable is like a very big Excel sheet spread across many computers.

Structure

```
Row Key + Column Family + Timestamp = Value
```

Diagram



Step-by-Step Working

1. Data is stored in rows.
2. Each row has a unique row key.
3. Columns are grouped into column families.
4. Timestamp stores different versions of data.
5. Data is distributed across many servers.

Real-Life Analogy

Google Search stores billions of web pages.
Each page can be stored with URL as row key, page content as column, and time of update as timestamp.

Advantages

- Highly scalable
- Stores huge structured data
- Fast access
- Supports distributed storage
- Good for large systems

Disadvantages

- Complex to understand
- Not good for small applications
- Does not support full SQL like relational database

Applications

- Google Search
- Gmail
- Google Earth
- Web indexing
- Big Data storage

Important Keywords

Column-Oriented, Row Key, Column Family, Timestamp, Distributed Storage, Scalability

Conclusion

BigTable is a powerful Google storage system used for large structured data. It is important for cloud and Big Data applications.

---

4. Dynamo

Introduction

Dynamo is a distributed database developed by Amazon.
It is designed for high availability and fast response.

Definition

Dynamo is a distributed key-value storage system developed by Amazon to provide high availability and fault tolerance.

Why It Is Needed

Amazon shopping website must work all the time.
Even if some servers fail, users should still buy products.

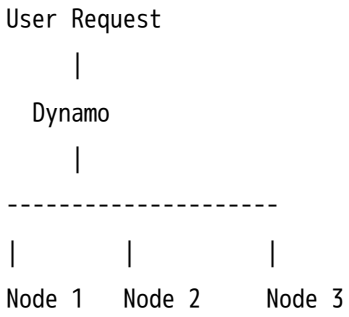
Easy Explanation

Dynamo stores data in key-value form.

Example:

```
ProductID101 → Mobile
ProductID102 → Laptop
```

Diagram



Step-by-Step Working

1. User sends request.
2. Dynamo finds key.
3. Data is searched from available nodes.
4. If one node fails, another node gives data.
5. System remains available.

Real-Life Analogy

If one shop branch is closed, customer can buy from another branch.

Same way Dynamo gives data from another server.

Advantages

- Highly available
- Fault tolerant
- Fast response
- No single point of failure
- Good for large online systems

Disadvantages

- Complex design

- Data consistency may be delayed
- Not suitable for relational data

Applications

- Amazon shopping cart
- E-commerce systems
- Online transactions
- Fast lookup systems

Important Keywords

Key-Value Store, High Availability, Fault Tolerance, Replication, Distributed Database, Eventual Consistency

Conclusion

Dynamo is a highly available distributed database used where system failure cannot be accepted, such as Amazon shopping systems.

---

5. Parallel Computing

Introduction

Parallel computing means doing many tasks at the same time.
It improves speed and performance.

Definition

Parallel computing is a computing technique in which multiple processors work together to solve a large problem faster.

Why It Is Needed

Large problems take too much time on one processor.

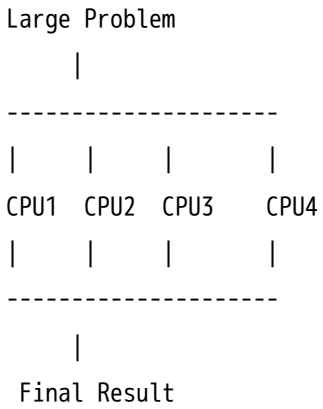
Parallel computing divides the problem into smaller parts.

Easy Explanation

One student writing full assignment takes more time.

If five students divide the work, assignment completes quickly.

Diagram



Step-by-Step Working

1. Big task is divided.
2. Each processor gets one part.
3. All processors work together.
4. Results are combined.
5. Final output is generated.

Advantages

- Faster execution
- Better CPU utilization
- Handles large data
- Improves performance

Disadvantages

- Complex programming
- Synchronization problem
- Hardware cost can be high

Applications

- Weather forecasting
- Big Data processing
- Artificial Intelligence
- Scientific research
- Cloud computing

Important Keywords

Multiple Processors, Simultaneous Execution, Speedup, Distributed System, Performance

Conclusion

Parallel computing is used to solve large problems faster by using many processors at the same time.

Memory Trick

```
Parallel = Many processors working together
```

---

6. HBase

Introduction

HBase is a distributed database used with Hadoop.

It is inspired by Google BigTable.

Definition

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

Why It Is Needed

HDFS stores data, but it is not good for fast random read/write.

HBase provides real-time access to Big Data.

Easy Explanation

HDFS is like a warehouse.

HBase is like a counter where you can quickly search and update items.

HBase Architecture

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

Components

Component	Work
Client	Sends request
HMaster	Manages region servers
Region Server	Stores and handles data
HDFS	Stores actual files

Step-by-Step Working

1. Client sends request.
2. HMaster manages the system.
3. Region Server handles data.
4. Data is stored in HDFS.
5. User gets fast read/write access.

Advantages

- Open-source
- Scalable
- Real-time access
- Good for huge data
- Column-oriented storage

Disadvantages

- Complex setup
- Needs Hadoop ecosystem
- Not suitable for small data
- Requires more memory

Applications

- Social media analytics
- Sensor data
- Banking logs
- Real-time Big Data search
- Recommendation systems

Important Keywords

HBase, HDFS, HMaster, Region Server, Column-Oriented, Real-Time Access

Conclusion

HBase is an open-source Big Data database built on HDFS. It is useful when fast access to large data is required.

---

7. Enterprise Batch Processing

Introduction

Batch processing means processing many records together at one time.
Enterprise batch processing is used by companies for large routine tasks.

Definition

Enterprise batch processing is the processing of large amounts of business data in groups or batches without continuous user interaction.

Why It Is Needed

Big companies process thousands or millions of records daily.
Doing this manually is impossible.

Easy Explanation

College result processing is a batch job.

All students’ marks are processed together and final results are generated.

Flow of Process

```
Collect Data
     ↓
Store Data
     ↓
Process in Batch
     ↓
Generate Output
     ↓
Reports / Results
```

Examples

- Salary processing
- Bank interest calculation
- Electricity bill generation
- Exam result generation
- Monthly sales reports

Step-by-Step Working

1. Data is collected.
2. Data is stored in system.
3. Batch job starts at scheduled time.
4. System processes all records.
5. Final reports are created.

Advantages

- Handles huge data
- Saves time
- Reduces manual work
- Good for repeated tasks
- Efficient for organizations

Disadvantages

- Not real-time
- Errors affect large data
- Delay in output
- Needs proper scheduling

Applications

- Banking
- Payroll systems
- Billing systems
- Insurance
- University result processing

Important Keywords

Batch Job, Bulk Data, Scheduled Processing, Enterprise System, Reports, Automation

Conclusion

Enterprise batch processing is useful for processing large business data together. It saves time and reduces human effort.

---

8. Relational Operations in Map Reduce

Introduction

Relational operations are database operations like selection, projection, join, and group by.

Map Reduce can perform these operations on huge datasets.

Definition

Relational operations in Map Reduce are database-style operations performed on large distributed datasets using map and reduce functions.

Why It Is Needed

Traditional databases become slow for huge data.

Map Reduce performs relational operations in parallel.

Main Relational Operations

Operation	Meaning
Selection	Select required rows
Projection	Select required columns
Join	Combine two tables
Group By	Group similar records
Aggregation	Sum, count, average

Easy Explanation

Suppose a college has data of 1 lakh students.

You want only CSE students. This is selection.

You want only name and roll number. This is projection.

Diagram

```
Large Table
    |
 Map Phase
    |
Filter / Select / Join
    |
Reduce Phase
    |
Final Result
```

Step-by-Step Working

1. Input table is divided.
2. Map function reads records.
3. Required operation is applied.
4. Intermediate data is grouped.
5. Reduce function combines results.
6. Final answer is produced.

Example: Selection

Find students with marks greater than 75.

Map checks each record:

```
If marks > 75 → emit record
```

Reduce combines all selected records.

Advantages

- Handles large database operations
- Faster than single-machine processing
- Useful for Big Data analytics
- Supports parallel execution

Disadvantages

- Complex for joins
- Not ideal for small databases
- More coding required

Applications

- Data warehousing
- Business analytics
- Student result analysis
- Banking transaction analysis

Important Keywords

Selection, Projection, Join, Group By, Aggregation, Key-Value Pair, Distributed Query

Conclusion

Relational operations in Map Reduce allow database-like processing on huge distributed datasets.

---

9. Parallel Efficiency

Introduction

Parallel efficiency shows how effectively multiple processors are used.
It tells whether parallel computing is giving good performance or not.

Definition

Parallel efficiency is the ratio of speedup to the number of processors used in parallel computing.

Formula

```
Parallel Efficiency = Speedup / Number of Processors
```

Where:

```
Speedup = Time taken by one processor / Time taken by multiple processors
```

Easy Explanation

If 10 processors are used, ideally work should become 10 times faster.
But due to communication and delay, actual speed may be less.

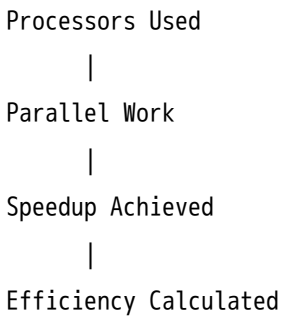
Example

If speedup = 8 and processors = 10:

```
Efficiency = 8 / 10 = 0.8 = 80%
```

This means processors are used 80% efficiently.

Diagram



Why Efficiency Decreases

- Communication delay
- Data transfer time
- Waiting time
- Unequal task distribution
- Synchronization problem

Advantages

- Measures system performance
- Helps improve resource usage
- Useful in Map Reduce analysis
- Shows processor utilization

Disadvantages

- Difficult to calculate for complex systems
- Efficiency depends on many factors
- More processors do not always mean more speed

Applications

- Map Reduce performance analysis
- Supercomputing
- Cloud computing
- Big Data processing

Important Keywords

Speedup, Processor Utilization, Scalability, Synchronization, Communication Overhead

Conclusion

Parallel efficiency helps measure how well processors are used in parallel systems. High efficiency means better performance.

🔥 Most Important 7-Mark Questions

1. Explain Map Reduce working with diagram.
2. Differentiate between GFS and HDFS.
3. Explain BigTable in cloud computing.
4. Explain Dynamo database.
5. What is parallel computing? Explain with example.
6. Explain HBase architecture.
7. Explain enterprise batch processing.
8. Explain relational operations in Map Reduce.
9. Explain parallel efficiency with formula.

⭐ Most Important 14-Mark Questions

1. Explain Map Reduce model in detail with working, diagram, advantages and applications.

2. Compare GFS and HDFS in detail. Which one is better and why?
3. Explain BigTable, HBase and Dynamo in detail.
4. Explain parallel computing and parallel efficiency in Map Reduce.
5. Explain relational operations and enterprise batch processing using Map Reduce.

🎯 PYQ-Based Expected Questions

Very High Probability

- Map Reduce working
- GFS vs HDFS
- HDFS / GFS architecture
- BigTable
- Dynamo

High Probability

- HBase
- Parallel computing
- Enterprise batch processing

Medium Probability

- Relational operations
- Parallel efficiency
- Map Reduce applications

⚡ One-Night Revision Notes

```
Map Reduce = Divide + Process + Combine

Map = small task processing
Reduce = combine final result

GFS = Google File System
HDFS = Hadoop Distributed File System

BigTable = Google column database
HBase = Open-source BigTable on HDFS
Dynamo = Amazon key-value database

Parallel Computing = Many processors together
Parallel Efficiency = Speedup / Number of processors

Batch Processing = Large data processed together
Relational Operations = Selection, Projection, Join, Group By
```

📌 Smart Study Plan

If only 2 hours left

1. Map Reduce – 40 min
2. GFS vs HDFS – 30 min
3. BigTable + HBase + Dynamo – 30 min
4. Parallel Efficiency + Batch Processing – 20 min

If 5 hours left

1. Map Reduce full answer – 1 hour
2. GFS vs HDFS – 1 hour
3. BigTable + HBase – 1 hour
4. Dynamo + Parallel Computing – 1 hour
5. Revision + diagrams – 1 hour

One-Night Priority Order

1. Map Reduce
2. GFS vs HDFS
3. BigTable
4. HBase
5. Dynamo
6. Parallel Computing
7. Batch Processing
8. Relational Operations
9. Parallel Efficiency

🧠 Memory Tricks

```
MAP = Make small parts
REDUCE = Rejoin results
```

```
GFS = Google File System
HDFS = Hadoop File System
```

```
BigTable = Big Google Table
HBase = Hadoop BigTable
Dynamo = Amazon Always Available
```

```
Parallel = Many processors
Efficiency = How well processors work
```

🚀 Topper Answer Writing Format

For any 7-mark or 14-mark answer, write in this order:

1. Definition
2. Introduction
3. Need / Importance
4. Diagram
5. Working
6. Example
7. Advantages
8. Disadvantages
9. Applications
10. Conclusion

Underline these keywords:

Distributed System, Scalability, Fault Tolerance, Replication, Parallel Processing, Key-Value Pair, Big Data, Map Phase, Reduce Phase, Metadata

This format will help you fill pages and score better marks.