Name - Shivam Mot
Class - TE (IT) T2
Roll no - 3025
Sub - DSBDA
Date - 16/4/25

17/20 Mail

Date : 16/4/25

* Theory Assignment - 1 *

1. Explain GFS along with diagram.

⇒

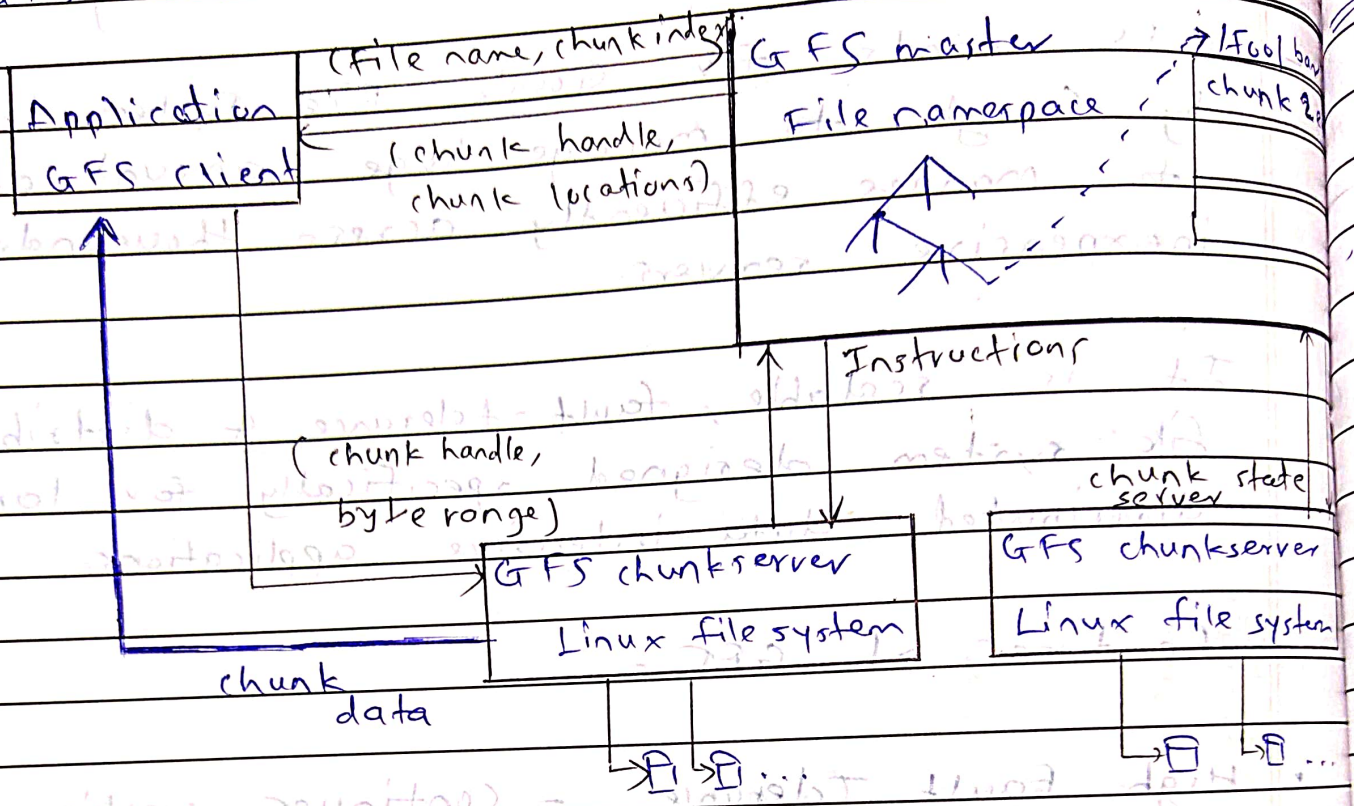
- GFS stands for Google File System, developed by Google to manage huge amount of data to manage efficiently across thousands of inexpensive servers.

- It is scalable, fault-tolerance, & distributed file system designed specifically for large distributed data-intensive applications.

* Features of GFT -

- High Fault Tolerance - continues working even if hardware fails frequently.
- High Throughput - optimized for large data reads/writes rather than low-latency small file access.
- Chunk-based storage - Files are divided into fixed-size chunks (typically 64MB), each identified by unique Chunk Handle.
- Master-Slave Architecture - Single master server manager metadata, & Chunk Server store actual data.
- Replication - Each chunk is replicated (default 3 copies) across different chunk server.

* GFS Architecture -



* components

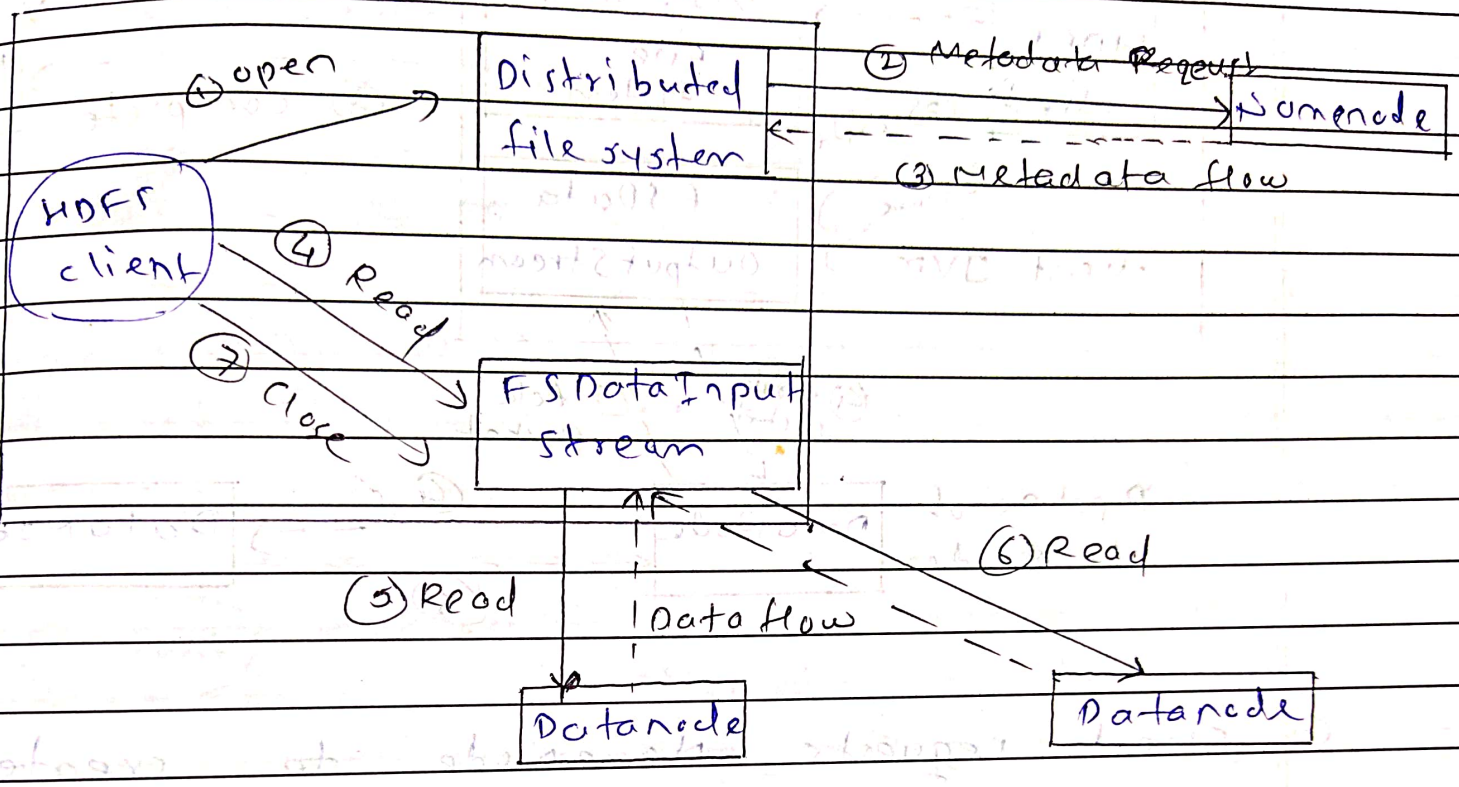
(1) **client** - Sends read & write requests. - communicates with master & stores chunk server

(2) **master server** - maintains metadata (namespace, mapping of files to chunks, locations of

(3) **chunk server** - store actual file data in chunks. Each chunk is stored on multiple servers.

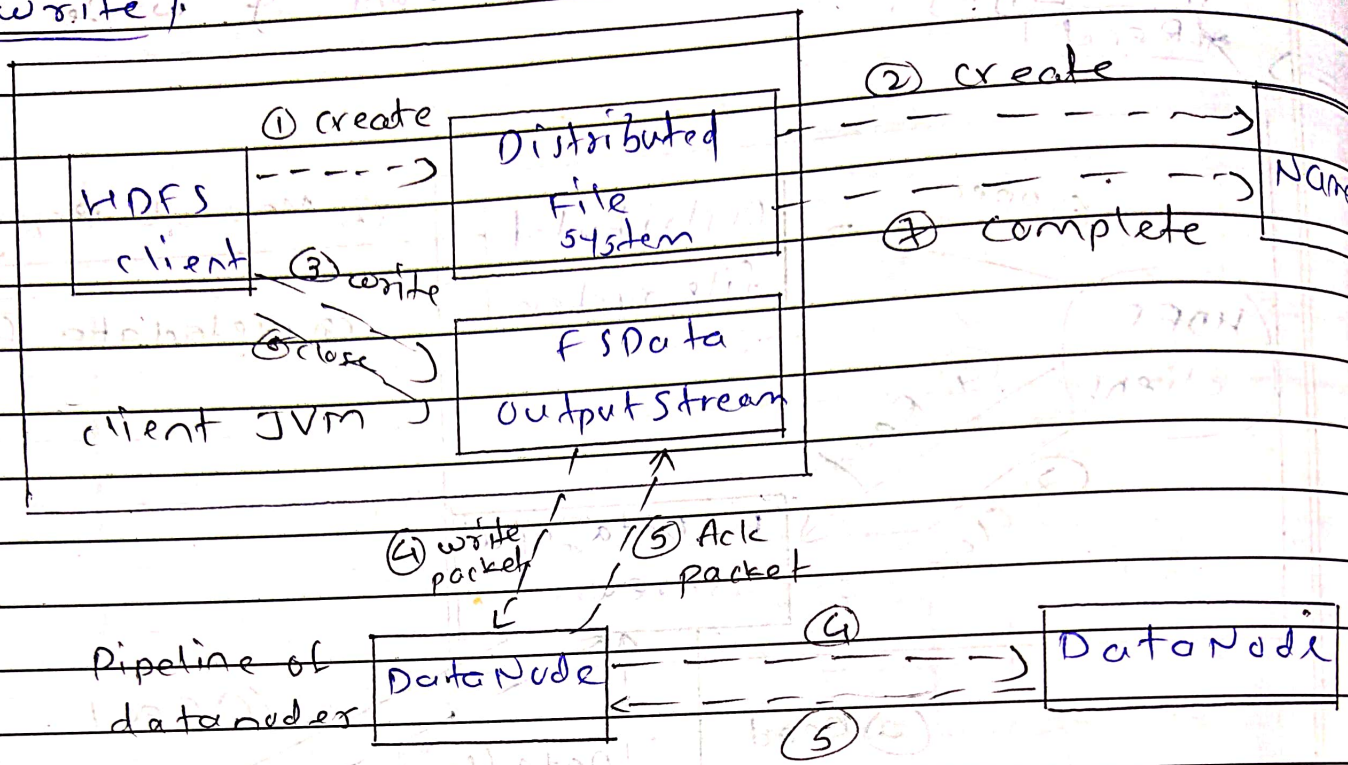
Explain in detail Read & write operations in HDFS.

Read



client contact NameNode to request location of file blocks (metadata only).
NameNode responds with list of DataNodes holding replicas of each block of file
client directly contacts nearest DataNode
Data is read block by block from DataNode & assembled into complete file on client side.
If DataNode - unavailable - client automatically switches to another replica provided by NameNode.

write:



① Client requests NameNode to create a new file entry.

② NameNode checks whether file already exists.

③ Client divides file into blocks. (default 128 MB).

④ NameNode provides list of DataNodes for each block where replicas will be stored (typically 3 replicas)

⑤ Client writes first block to first DataNode.

1) DataNode forwards block to second DataNode.

2) second DataNode forwards it to 3rd DataNode. (pipelining).

Date :

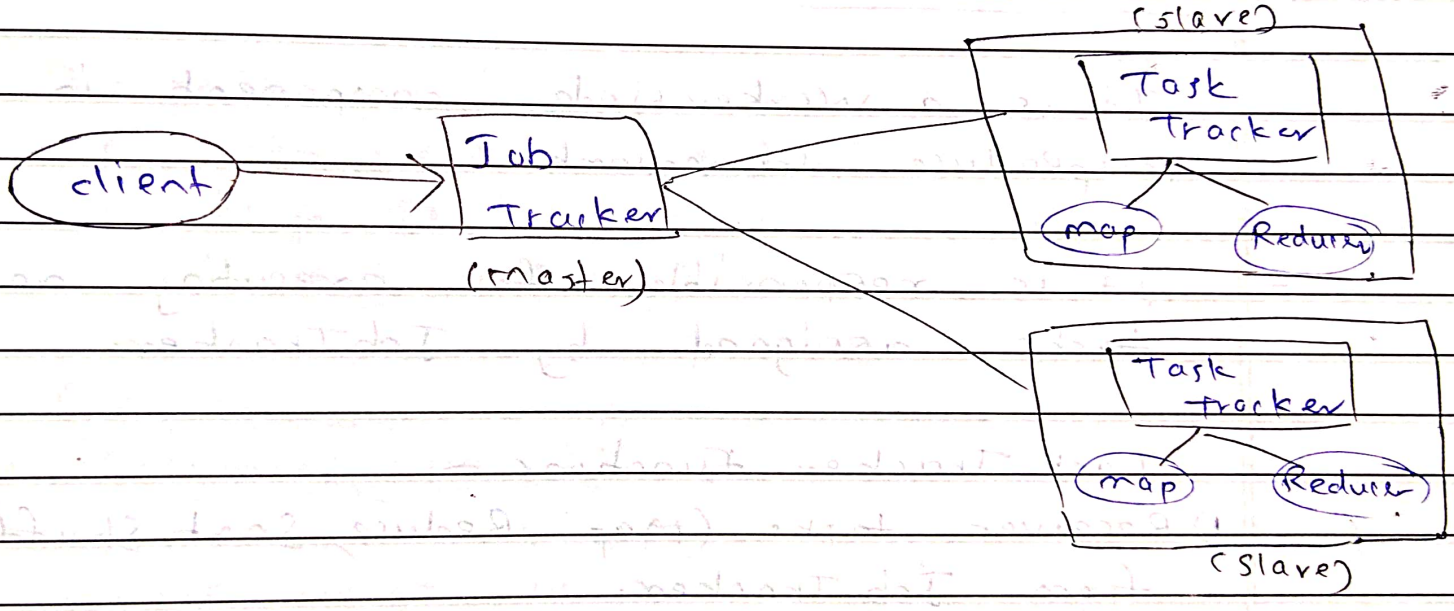
⑥ After successful writing of replication of each block, acknowledgment is sent back through pipeline to client.

⑦ Once all blocks are written & acknowledged, NameNode updates its metadata.

Note - Write Once, Read many times.

a) Explain with Diagram -
① Job Tracker ② Task Tracker

⇒



① Job Tracker

- Master node component in MapReduce framework.
- It is responsible for managing resources & coordinating execution of MapReduce jobs across cluster.

Date :

b) (3) No SQL databases.

=>

- NoSQL - "Not Only SQL"
- It provide a way to store & retrieve data that is different from traditional RDBMS.
- They are designed for large volumes of data, unstructured data, scalability.
- Characteristics -
  - Schema-less
  - High-Performance
  - Variety of Data Models.
  - Horizontal Scalability
  - Flexibal Data types
- Types -

i) Document-based

- Stores data in JSON-like documents. Best for nested data.
- e.g. - MongoDB, CouchDB
- e.g. - document - data stored -

```
{
    "id" : 101,
    "name" : "Shubham",
    "courses" : ["ML", "DSA"],
}
```

- No strict schema.

2) Key-value Store :-

- Data stored as key-value pairs.
- Simple & fast for basic lookup op
- e.g.- Redis, DynamoDB
- e.g.- Data stored like

key	value
user:1	{"name": "Om", "age": 25
session:123	{"token": "abcd"}

- ideal for caching, session management

3) Column-based

- Data stored as ~~key-value~~ in columns rather than rows.
- Efficient for large analytical queri
- columns grouped into familier
- e.g. Apache cassendra, HBase
- e.g. Data stored like -

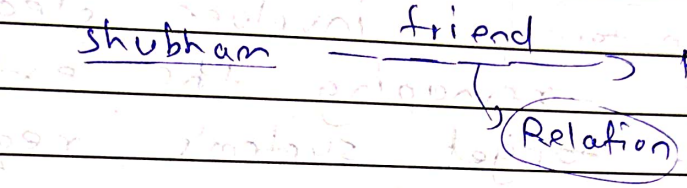
Id	Name		Course
	fname	lname	
1	Shub	Khot	ML
2	Jay	-	DS
3	Om	-	DS

- Good for fast retrieval applicati & massive datasets.

Date :

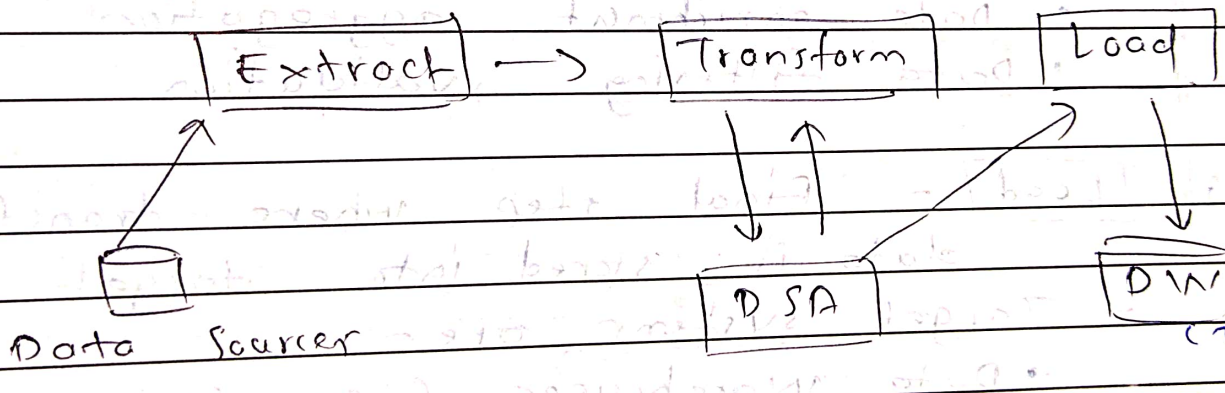
4) Graph-based

- Data stored as node & edges
- Designed to store relationships b/w entities
- Neo 4J, Amazon Neptune
- e.g.



4. Explain ETL in detail

=>



- ETL (Extract, Transform, Load) is critical step in building consistent data pipelines for analytics & BI.

① Extract

- Process of gathering data from various source systems
- Sources can be —
  - Databases (SQL) (NoSQL)

Date :

- APIs
- Spreadsheets
- Cloud Storage
- Logs & sensor data

– Goal is to collect raw data without losing info.

(2) **Transform** – involves cleaning, modifying, reshaping the extracted data to match target system's requirement

– tasks performed –
- Data cleaning (removing errors, duplicates)
- Data Standardization
- Data enrichment, aggregation
- Data filtering, validation

(3) **Load** – Final step where transformed data is stored into target system.

– Target systems are –
- Data Warehouses (Snowflake, Amazon Redshift)
- Databases (Postgre, MySQL)
- Data lakes (AWS S3, Azure Data Lake)

– Some ETL Tools examples –
1. Apache NiFi, Talend Open Studio
2. AWS Glue, Google cloud DataFlow
3. IBM Data storage..