Hadoop.

Q1) List the use and responsibilities of following tools of Hadoop Ecosystem.

i) HDFS: (Hadoop Distributed file system) is the primary storage system used by hadoop to store large data sets across multiple machines in a distributed fashion.

responsibilities.

1) storage of large datasets: HDFS splits large datasets into blocks and stores them across different nodes for parallel processing.

2) Fault tolerance: HDFS replicates data across multiple nodes to ensure data is not lost in case of hardware failure.

3) Data Accessibility: It allows seamless access to stored data.

4) High Throughput: It is high throughput rather than low latency for access to large datasets, making it ideal for batch processing.

5) Scalability: HDFS handles datasets that grow exponentially.

ii) YARN (Yet Another Resource Negotiator) is Hadoop's resource management system that handles job scheduling and resource allocation across the Hadoop cluster. It allows multiple applications to run simultaneously by dynamically managing resources.

Responsibilities:

1) Resource Management: allocates resources like CPU and memory to various applications running on the Hadoop cluster.

2) Job scheduling: It schedules jobs efficiently, managing the execution of tasks based on available resources.

3) Multi tenancy: enables running different types of applications on the same cluster without conflict.

4) Fault tolerance: YARN tracks & restarts failed jobs, ensuring smooth execution of applications.

5) Dynamic Resource Allocation: It ensures that resources are assigned to applications based on their current demands, improving cluster utilization.

iii) FLUME: is a distributed, reliable and available service for efficiently collecting, aggregating, and moving large amount of log data or streaming data from various sources into centralized storage like HDFS.

Responsibilities.

1) Data Ingestion: Flume helps in ingesting data from multiple sources into HDFS

2) Streaming Data: It is optimized for continuous & real time streams, handling large volumes of event data.

3) Fault tolerance: Flume provides guaranteed delivery of data, even in case of node failure

4) Scalability: can be scaled horizontally by adding more agents to handle

5) Customizability: Flume supports various custom sources, sinks & channel types to meet different data ingestion needs.

iv) Sqoop is used for transferring data b/w Hadoop & RDBMS. It imports data from databases like MySQL, oracle & postgre SQL

into HDFS and exports processed data back to the databases
Responsibilities.
i) Data Import: Sqoop imports structured data from relational databases into Hadoop for further analysis.
ii) Data Export: It exports processed data from HDFS back to the relational database for reporting or further use.
iii) Efficient Data transfer: Sqoop uses parallel data transfer to the relational database providing to efficiently transfer large datasets to and from Hadoop, ensuring high performance
iv) Data transformation: During Import/export, it can also transform the data into a desired format.
v) Schema Mapping: Automatically generates schema for importing tables ensuring smooth integration with relational databases.

e) Zookeeper: is a centralized service for maintaing configuration information, naming, providing distributed synchronization and group services within a Hadoop cluster. It is essential for coordinating distributed applications.
Responsibilities
i) Coordination: coordinates various distributed services & applications in a cluster to ensure they operate synchronously.
ii) Configuration Management: Zookeeper stores & manages configuration information and distributes it across all nodes in the cluster.
iii) Leader Election: facilitates leader election in a distributed system to ensure one master node control task execution while others are standby.
iv) Distributed Locking: provides a locking mechanism to avoid conflicts in accessing shared resources in a distributed environment.
v) Fault tolerance: Zookeeper itself is highly available and ensures consistency across nodes even if one or more nodes fail.

f) Hive: is a data warehouse infrastructure built on top of Hadoop that provides data summarization, querying, and analysis through an SQL like interface (HiveQL), it allows users familiar with SQL to query data stored in HDFS without writing complex MapReduce programs.
Responsibilities:
i) SQL like querying: Hive allows users to write queries in HiveQL to retrieve and manipulate large datasets in HDFS.
ii) Data warehousing: Hive is primarily used for data warehousing tasks like data summarization, ad hoc querying, analysis
iii) Schema on Read: Hive applies a schema to data as it is read (rather than when it is written), making it flexible for analysing unstructured or semi structured data.
iv) Integration with Hadoop: Hive translates HiveQL queries into mapreduce jobs, allowing data to be processed on Hadoop's distributed computing framework.
v) Extensibility: Hive supports user defined functions (UDFS) for custom data processing & analysis, enabling users to extend its capabilities.

PAGE NO.
DATE / /

Q(2) Explain the responsibilities of name node and data nodes in HDFS, Differentiate normal file system with HDFS in tabular column.

A) The Name Node is the central part of HDFS and acts as the master server responsible for managing the metadata and file system namespace. it coordinates the storage & retrieval of the data blocks from the data nodes

Responsibilities of Name Node

i) Metadata Storage: Name Node stores metadata for the entire file system in memory.

ii) Namespace Management: It maintains the hierarchical structure of the file system, including directories, subdirectories, and files.

iii) Data block replication: To ensure data is fault tolerant by replicating blocks across multiple data Nodes according to configured replication factor.

iv) Data blocks management: Name Node tracks the locations of data blocks stored on different data Nodes, ensuring they are correctly replicated & available.

v) client Interaction: When a client wants to read or write data, the Name Node provides the necessary block info and coordinates access to data Nodes.

vi) Failover support: In case of data Nodes failures, the name node ensures that the data is available by tracking replicas & initiating replication when necessary.

Data Nodes serve as the workers in HDFS, responsible for storing actual data blocks. they regularly communicate with the Name Node to ensure the system functions

Responsibilities of Data Nodes:

i) Storage of data blocks: Data Nodes store and manage the physical storage of data blocks assigned by the Name Node

ii) Data Blocks reports: They periodically send block reports to the Name node, indicating the blocks they are storing

iii) Data replication: Data Nodes participate in the replication process by replicating data to other Data Nodes as instructed by the Name Node.

iv) Data Read/write: They manage the actual data I/O operations providing data to clients or accepting new data for storage

v) Heartbeat Signals: Data Nodes send heartbeat signals to the Name Node at regular intervals to indicate they are functioning properly.

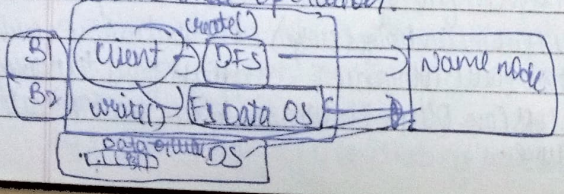
vi) Error Handling: If a data node fails, its data blocks are replicated on other nodes by the name node to ensure fault tolerance.

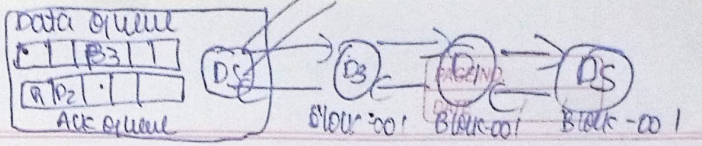
Difference between Normal file system and HDFS

	Normal file system	HDFS (Hadoop Distributed)
Storage	Single machine (centralized storge)	Distributed across multiple.
Fault Tolerance	limited fault tolerance; data loss can occur on hardware failure.	High fault tolerance: data blocks are replicated.

Data size	Handles small to medium sized files.	Designed to handle very large datasets (GB, TB, PB)
Handling	local file system, accessed by one machine	distributed file system, accessible across the cluster.
Block size	small block size (4KB, 8KB)	large block size (128 MB or 256 MB)
File system type	Hierarchical file system for general purpose use.	Designed for write once, read many access pattern.
Replication	No automatic replication, backup needed manually	Automatic replication of data blocks for fault tolerance
Scalability	limited scalability as it is restricted to single machine storage capacity	High scalability, can easily scale out by adding more data nodes.
Data processing	data processing is done on the same machine where the file system resides	data is processed in a distributed manner across multiple nodes.
Metadata	stores metadata with handling each file locally	Name node stores metadata in memory while data nodes store only data.

Q3) With diagram show how write operation is performed in HDFS. Explain the steps to recover from failure during HDFS write operation.





i) Client request: Sends a request to Name Node to create a file & write data. The Name Node checks if the file already exists and whether the client has the appropriate permissions to create the file.

ii) Block Assignment: The Name Node splits the file into blocks (128 MB) & identifies a list of Data Nodes to store each block. It returns the list Data Nodes where the replicas of the data blocks should be stored.

iii) Pipeline Setup: The client establishes a pipeline connection with the first Data Node (Replica 1). This Data Node, in turn, communicates with the second Data Node (Replica 2), and the second Data Node communicates with the third Data Node. This is done in a pipline fashion, ensuring all replicas are written simultaneously.

iv) Acknowledgment: After the block is written to all replicas the Data Nodes acknowledge the successful write.

v) Writing Data: The client starts writing data blocks. Each block is written to Data Node.

vi) File Close: After all blocks are written & acknowledged the client informs the Name Node that the file has been closed. The Name Node then commits the file creation.

Steps to recover from failure during HDFS write operation

1) Data Node failure: If Node fails.
The client is notified about the failure.

The name node removes the failed data node & rearranges the write.
* The pipeline resumes.

2) Client failure
* Name node waits for the client to reestablish the connection and continue the write operation.
* If the client does not reconnect, the name node marks the file as incomplete & removes written blocks.

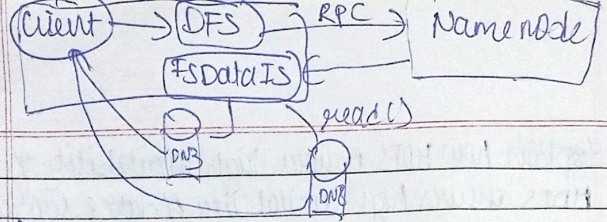
3) network failure
* The current block being written is saved in the existing data nodes that have received the data.
* Once the network recovers, the name node reestablishes the connection with new data nodes.

4) Name node failure:
* HDFS uses a secondary name node or checkpoint node to ensure metadata and file system state.
* The system brings the name node back online, and the incomplete write operations can be restarted from where they left off.

5) Handling data corruption uses checksums to verify data integrity.
Name node detects this using block reports.
The corrupt block is deleted and a new replica is created from a healthy copy on another data node.

Q4) Explain how HDFS ensures high availability of data & services.
HDFS ensures high availability of data & services through a combination of redundancy, fault tolerance, and failover mechanisms.
i) Data Replication: Each block is replicated 3 times.
* primary replica, secondary replica, tertiary replica
ii) Fault tolerance: If a node fails, the name node automatically redirects the client to another node.
* self healing mechanism.
iii) Name node High availability. Traditionally HDFS had a single name node, creating a single point of failure. However modern Hadoop version implement name node High Availability (HA).
* Active and standby name node, failover mechanism
iv) Rack Awareness: To ensure replicas are placed on different rack.
v) Heart beat & Block report.
vi) Checksum verification.
vii) Backup & Recovery Mechanism.

Q5) With diagram show how read operation is performed in HDFS. Explain the steps involved in read operation.
A) The read operation in HDFS involves interactions b/w the client, name node, data nodes.



i) Client Request to Name node
ii) block location: Retrieval from Name node.
iii) connecting to data nodes.
iv) Data Block Reading.
v) Block replicas as backup.
vi) Data Return to client
vii) No Interaction with name node During data transfer.

Q6) Explain the terms heart beat, edit log and check points

1) heartbeat: periodic signal sent by a Data node to the name node to indicate that is alive & functioning correctly.
- Health monitoring, Data node availability.
- Mechanism: Data nodes send heart beats at regular intervals (default is every 3 seconds.

2) edit log: is a file maintained by the name node that records all changes made to the file system metadata. It logs operations such as file creation, deletion and modifications

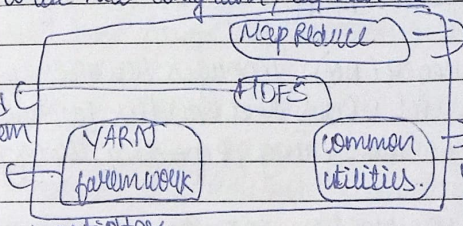
3) check point:
is a snapshot of the file system metadata and its state, including the edits recorded in the edit log.

the checkpoint process consolidates the in-memory metadata and the edit log into a single file system image

PAGE NO.
DATE / /

metdata Recovery, efficiency.

Q7) write neat diagram, explain the architecture of HDFS



Client interacts with HDFS to perform read & write.
HDFS → Name node (master node)
Data node (slave node).
blocks (Data unit of storage in HDFS (128MB)
Map reduce → blocks of data are Mapped.
and shuffle & sort then reduced by reducers.
RYARN → Resource Negotiator & Job Scheduling Resource Management.

Q8) List the major components of YARN. Explain the responsibilities of YARN.
YARN (yet another resource Negotiator) is a resource management layer of Hadoop that allows for more flexible & efficient resource management compared to the other Map reduce framework

Resource Manager allocates cluster resources.
* keeps track of available resources
* scheduler, Application management.

Node Manager (NM): manages resources on individual nodes and handles container execution
* Resource monitoring, reporting, log aggregation

Application Master: manages the lifecycle of a specific application & coordinates with the resource manager & NodeManager.

Responsibilities of YARN
* Resource allocation, Application scheduling
Resource monitoring, fault tolerance & recovery
Cluster utilization, Scalability

Q9) list out the responsibilities of Resource Manager
* Resource allocation & Management.
* Application submission & Management
* Application Master coordination
* Resource monitoring & Reporting
* failure Detection & Recovery
* release management * scalability
load balancing * integration with other components.

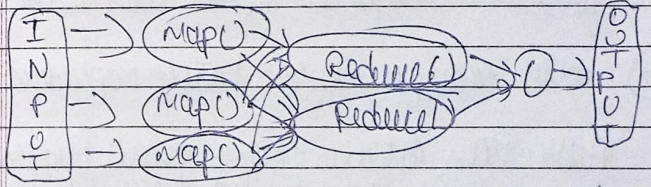
Q10) Explain different schedulers used in YARN
i) capacity scheduler ensures that each queue get its fair share of resources and that large applications do not monopolize cluster.
ii) fair scheduler aims to allocate resources equally among running applications to ensure that all applications receive fair share
iii) FIFO is the simplest form of scheduling in YARN

Q11) Explain different types of failure in mapreduce jobs.
i) Task failure occur when an individual map or reduce task fails to compete successfully.
ii) Node failure occur when a node in the hadoop cluster becomes unresponsive or crashes.
iii) Resource Manager failure: Can disrupt the management of job scheduling & resource allocation
iv) Data Node failure occur when a DataNode storing the data blocks becomes unresponsive
v) Application Master failure occur when the application Master, which manages the lifecycle of jobs
vi) Network failure can cause communication issues b/w nodes, leading to task failures or delay

12) Give different reasons for task failure in map reduce job give the steps for recovering from Application master failure.
i) Hardfailure * Disk failure * Memory Issues
ii) Software Bugs: Application Bugs * library or Dependency Issues.
iii) Data Corruption: Corrupted Input Data data format Issues.
iv) Resource Constraints - CPU or memory limitations, Insufficient Disk Space.
v) Network Issues: network partitions, Slow Network.
vi) Configuration Errors: Misconfiguration Resource misallocation.
vii) Dependency failure: external service failure
viii) file System Issues: HDFS failure.

Steps for Recovering from Application Master
i) Detect failure ii) Restart Application Master iii) Recover Application state iv) Handle Data and Resource Reallocation v) Notify & Update users vi) Post failure Actions.

Q13) With neat diagram, show how map reduce jobs are executed in Hadoop environment.
* Job submission: user submits job.
* Job Initialization: creates a job context
* Job scheduling: schedules the job's tasks (map and reduce) across the cluster nodes.
* Map Task execution: input split & applies map()
* Shuffle & sort: transfer the sorted map output to the appropriate reduce tasks.
* Reduce task execution: process intermediate key-value pairs, applies the reduce function
* Job completion - Job tracker & Resource manager * Job status.



Q14) Give the responsibilities of
a) Application Master i) Application Manager
b) Node Manager d) Resource Manager
a) Application Master: Application life cycle management, Resource request & Negotiation Task Scheduling, Application Monitoring

iii) Application Manager manages the application master and acts a middle ware for the resource manager & Application Master. If any Application Master is failed then it restart, re execution, re compete that.

iv) Node Manager : Resource monitoring, container management, task execution, Health reporting & log Aggregation.

v) Resource manager : oversees resource allocation, scheduler management resource tracking application coordination, failure Management & job submission handling.

Q(5) with example, explain Hadoop mapreduce process

Public class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
String line = value.toString();
StringTokenizer itr = new StringTokenizer(line);
while (itr.hasMoreTokens()) {
word.set(itr.nextToken());
context.write(word, one);
}}}

public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values) {
sum += val.get();
}
result.set(sum);
context.write(key, result);
}
}

count the number of occurences of each word in a large collection of text files.
mapreduce process.
Job submission, Job initialization.
map phase, shuffle and sort phase, reduce phase, Job completion.