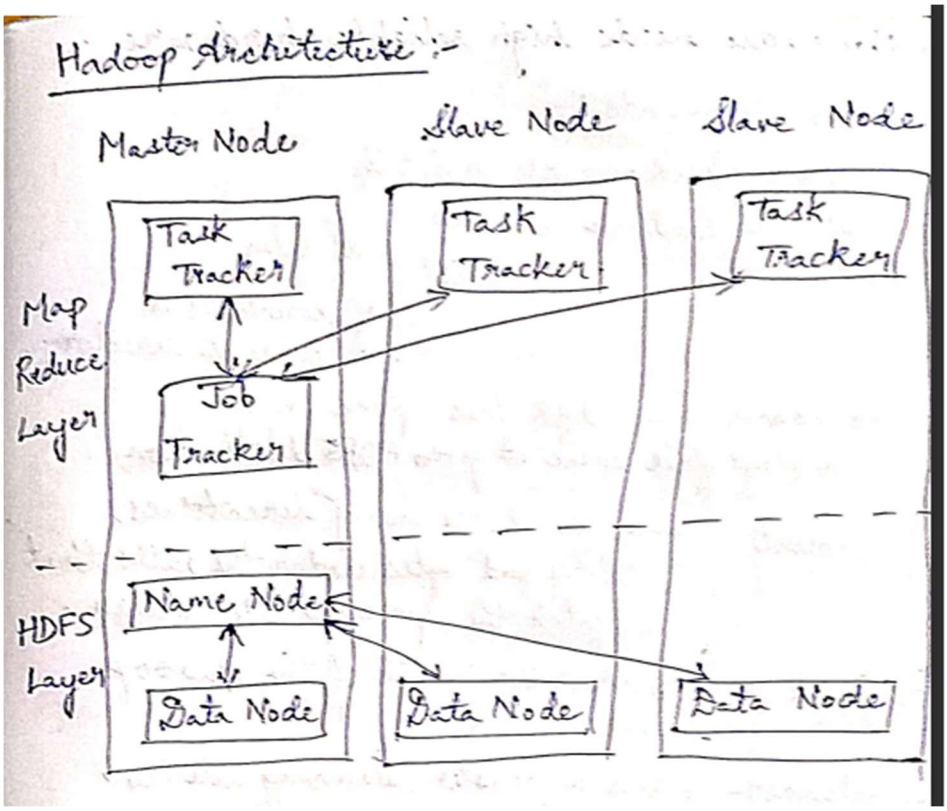
1)What is Hadoop? Explain Architecture of Hadoop.



Hadoop is an open-source software framework used for storing and processing Big Data in a distributed manner on large clusters of commodity hardware.

Hadoop Architecture:

The Hadoop Architecture Mainly consists of 4 components.

- MapReduce
- HDFS(Hadoop distributed File System)
- YARN(Yet Another Resource Negotiator)
- Common Utilities or Hadoop Common

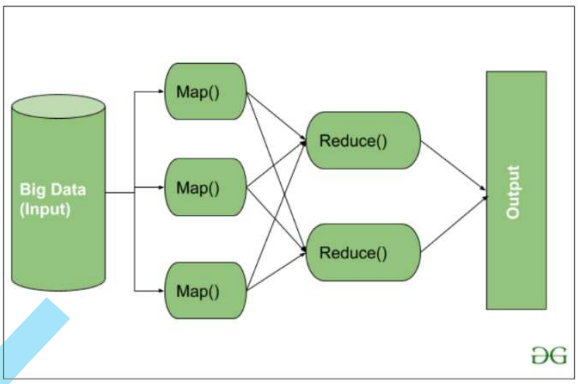
1. MapReduce

MapReduce nothing but just like an Algorithm or a data structure that is based on the YARN framework.

The major feature of MapReduce is to perform the distributed processing in parallel in a Hadoop cluster which Makes Hadoop working so fast.

MapReduce has mainly 2 tasks which are divided phase-wise:

In first phase, **Map** is utilized and in next phase **Reduce** is utilized.



Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples

Here, we can see that the *Input* is provided to the Map() function then it's *output* is used as an input to the Reduce function and after that, we receive our final output.

2. HDFS

HDFS in Hadoop provides Fault-tolerance and High availability to the storage layer and the other devices present in that Hadoop cluster. Data storage Nodes in HDFS.

- NameNode(Master)
- DataNode(Slave)

Hadoop comes with a distributed file system called HDFS. In HDFS data is distributed over several machines and replicated to ensure their durability to failure and high availability to parallel application.

Data in HDFS is always stored in terms of blocks.

NameNode:

- NameNode works as a Master in a Hadoop cluster that guides the Datanode(Slaves).
- Namenode is mainly used for storing the Metadata i.e. the data about the data. Meta Data can be the transaction logs that keep track of the user's activity in a Hadoop cluster.
- Meta Data can also be the name of the file, size, and the information about the location(Block number, Block ids) of Datanode that Namenode stores to find the closest DataNode for Faster Communication.
- Namenode instructs the DataNodes with the operation like delete, create, Replicate, etc.

DataNode:

- DataNodes works as a Slave.
- DataNodes are mainly utilized for storing the data in a Hadoop cluster, the number of DataNodes can be from 1 to 500 or even more than that. The more number of DataNode, the Hadoop cluster will be able to store more data. So it is advised that the DataNode should have High storing capacity to store a large number of file blocks.

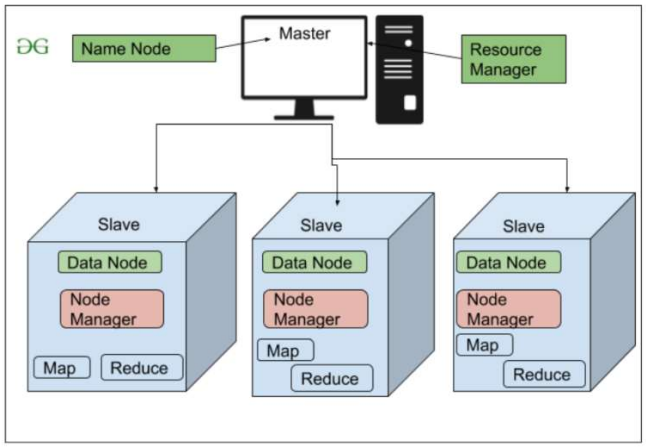
Secondary name node

Secondary NameNode in hadoop is a specially dedicated node in HDFS cluster whose main function is to take checkpoints of the file system metadata present on namenode. It is not a backup namenode. It just checkpoints namenode's file system namespace. The Secondary NameNode is a helper to the primary NameNode but not replace for primary namenode.

As the NameNode is the single point of failure in HDFS, if NameNode fails entire HDFS file system is lost. So in order to overcome this, Hadoop implemented Secondary NameNode whose main function is to store a copy of **FsImage** file and **edits** log file.

Secondary NameNode **is not a true backup Namenode** and cann't serve primary NameNode's operations.

High Level Architecture Of Hadoop



3. YARN(Yet Another Resource Negotiator)

YARN is a Framework on which MapReduce works.

YARN performs 2 operations that are Job scheduling and Resource Management.

The Purpose of Job schedular is to divide a big task into small jobs so that each job can be assigned to various slaves in a Hadoop cluster and Processing can be Maximized.

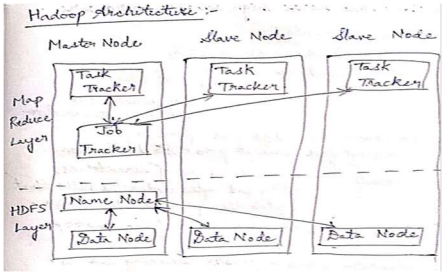
Job Scheduler also keeps track of which job is important, which job has more priority, dependencies between the jobs and all the other information like job timing, etc.

The use of Resource Manager is to manage all the resources that are made available for running a Hadoop cluster.

Features of YARN

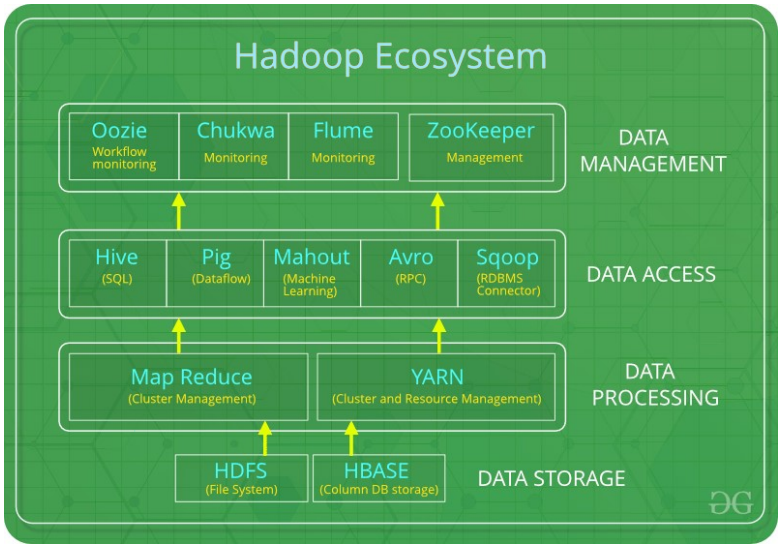
- Multi-Tenancy
- Scalability
- Cluster-Utilization
- Compatibility

4. Hadoop common or Common Utilities: Hadoop common or Common utilities are nothing but java library and java files. These utilities are used by HDFS, YARN, and MapReduce for running the cluster.



2)Write about Hadoop Ecosystem.

Hadoop Ecosystem:



The components that collectively form a Hadoop ecosystem:

- HDFS: Hadoop Distributed File System
- YARN: Yet Another Resource Negotiator
- MapReduce: Programming based Data Processing
- Spark: In-Memory data processing
- PIG, HIVE: Query based processing of data services
- HBase: NoSQL Database
- Mahout, Spark MLLib: Machine Learning algorithm libraries
- Solar, Lucene: Searching and Indexing
- Zookeeper: Managing cluster
- Oozie: Job Scheduling

The Flume is a service which helps in ingesting unstructured and semi-structured data into HDFS.

The major difference between Flume and Sqoop is that:

Flume only ingests unstructured data or semi-structured data into HDFS.
While Sqoop can import as well as export structured data from RDBMS or Enterprise data warehouses to HDFS or vice versa.

chukwa,flume?

HBase:
It supports all kinds of databases.
Helps to sort data, search and indexing is also fast

YARN:
Yet Another Resource Negotiator.
It helps to manage the resources across the cluster.
It helps in managing & scheduling the resource allocation.
Resource Manager, Application Manager and Node Manager — are its components.
Helps in allocation resources like CPU, memory...

Data access:

Hive:
Hive Query language performs reading & writing of large data sets.
Hive Command line helps in processing of queries.
It allows bash processing & real-time processing

Pig:
It is also query based lang. that is developed by Yahoo
It allows structuring the data flow, analyse & process the large datasets.

Mahout (ML):
It allows machine learnability to system.

AVRO:
It supports Remote Procedure call.

Apache's Path:
It is used to handle all processing tasks like bash processing, realtime processing, iterative processing.

Zoo Keeper:
It is management of coordination, synchronization.

Oozie: (Also for chukux and clume)
Job scheduler — reading, writting, closing

Lucene:
Searching and indexing.

Hive is a data warehouse and an ETL tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS).
HIVE is a interface between HDFS and User.

NOTE: HDFS, MapReduce and YARN already described above.

PIG:

Pig is a high-level platform or tool which is used to process the large datasets
Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.

- It is a platform for structuring the data flow, processing and analyzing huge data sets.
- Pig does the work of executing commands, After the processing, pig stores the result in HDFS.
- Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the JVM.

HIVE:

- HIVE performs reading and writing of large data sets. Its query language is called as HQL (Hive Query Language).
- It is highly scalable as it allows real-time processing and batch processing. All the SQL datatypes are supported by Hive.
- Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
- JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

Mahout:

- Mahout, allows Machine Learnability to a system or application.
- It provides various libraries or functionalities such as filtering, clustering, and classification. It allows invoking algorithms as per our need with the help of its own libraries.

Apache Spark:

It's a platform that handles all the processing tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.

- Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing.

Apache HBase:

It's a NoSQL database which supports all kinds of data. It provides capabilities of Google's BigTable, thus able to work on Big Data sets effectively.

- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data

Lucene:

Lucene perform the task of searching and indexing with the help of some java libraries, However, Lucene is driven by Solr.

Zookeeper:

Zookeeper overcame huge issue of management of coordination and synchronization among the resources of Hadoop by performing synchronization, inter-component based communication, grouping, and maintenance.

Oozie:

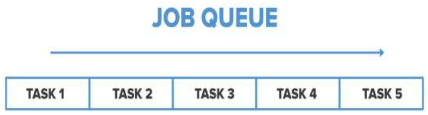
Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit.
There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs.

3)What is FIFO, Fair, Capacity Schedulers?

There are mainly 3 types of Schedulers in Hadoop:

1. FIFO (First In First Out) Scheduler.
2. Capacity Scheduler.
3. Fair Scheduler.

A **Job queue** is nothing but the collection of various tasks that we have received from our various clients.



1)FIFO Scheduler

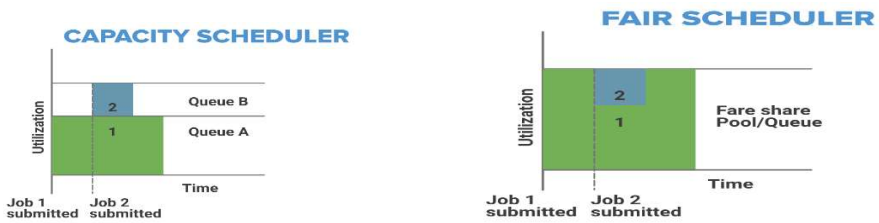
As the name suggests FIFO i.e. First In First Out, so the tasks that comes first will be served first. This is the default Scheduler we use in Hadoop. In this method, once the job is scheduled, no intervention is allowed. So sometimes the high-priority process has to wait for a long time since the priority of the task does not matter in this method.

2. Capacity Scheduler

In Capacity Scheduler we have multiple job queues for scheduling our tasks. Each job queue has it's own slots to perform its task.

In case we have tasks to perform in only one queue then the tasks of that queue can access the slots of other queues also as they are free to use.

The capacity Scheduler mainly contains 3 types of the queue that are root, parent, and leaf



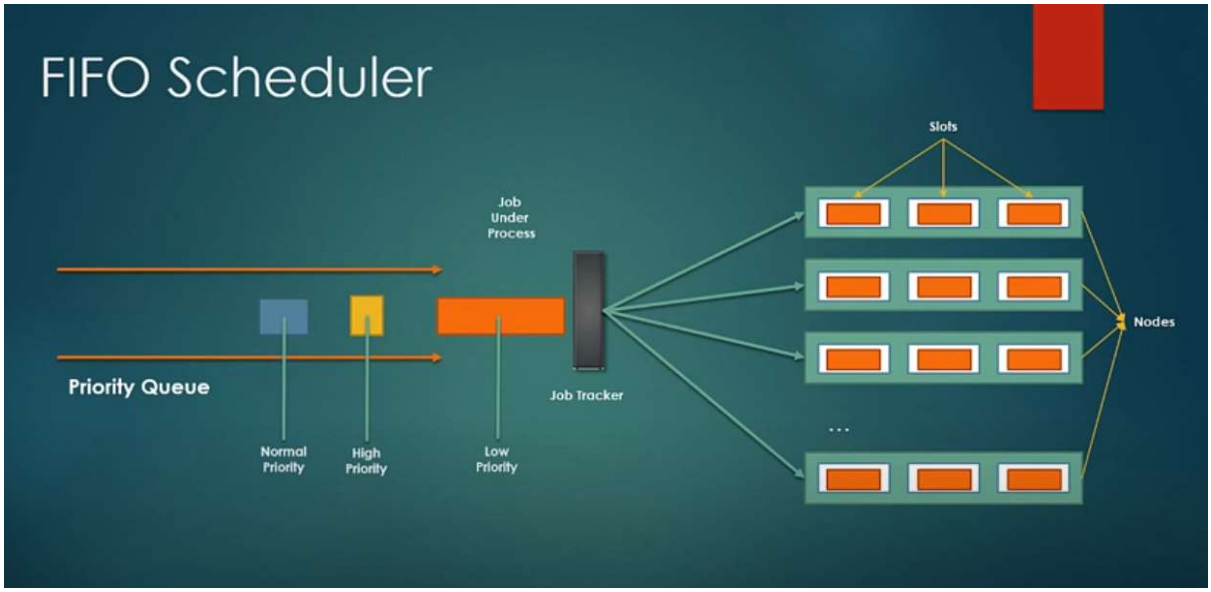
3. Fair Scheduler

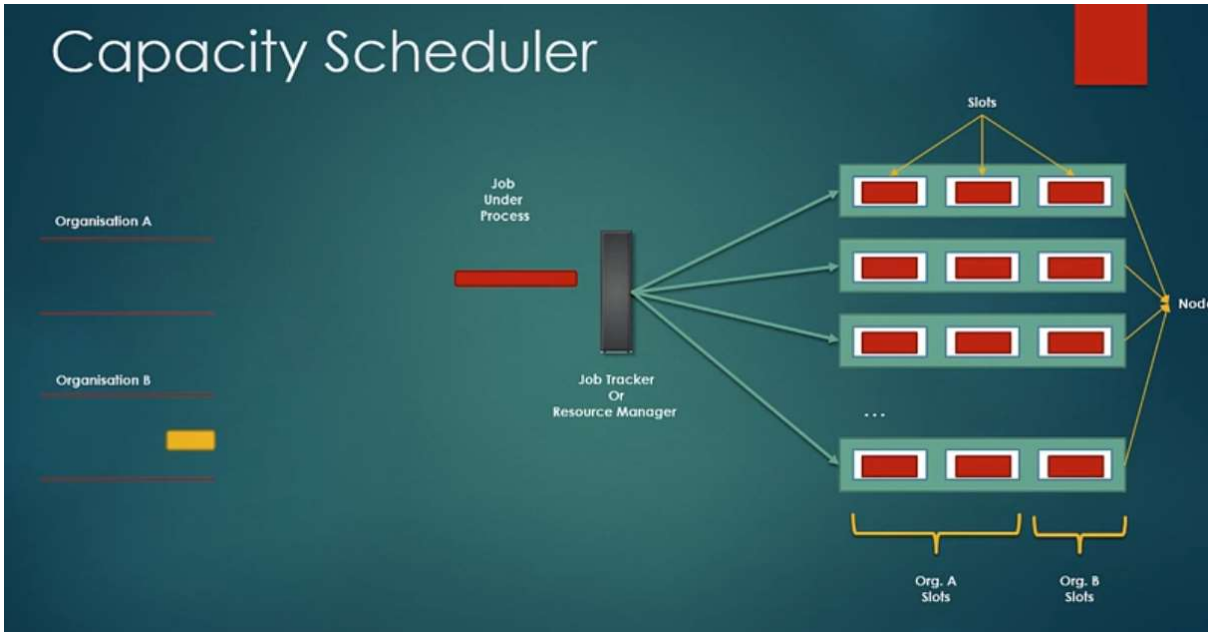
The Fair Scheduler is very much similar to that of the capacity scheduler.

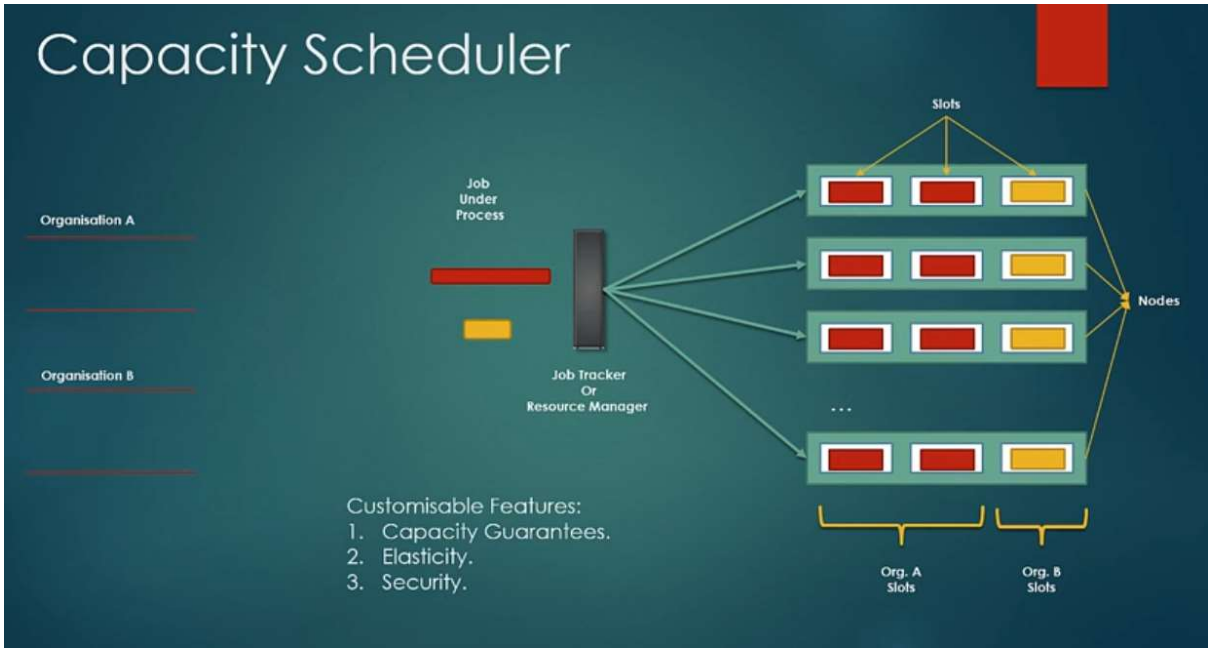
The priority of the job is kept in consideration.

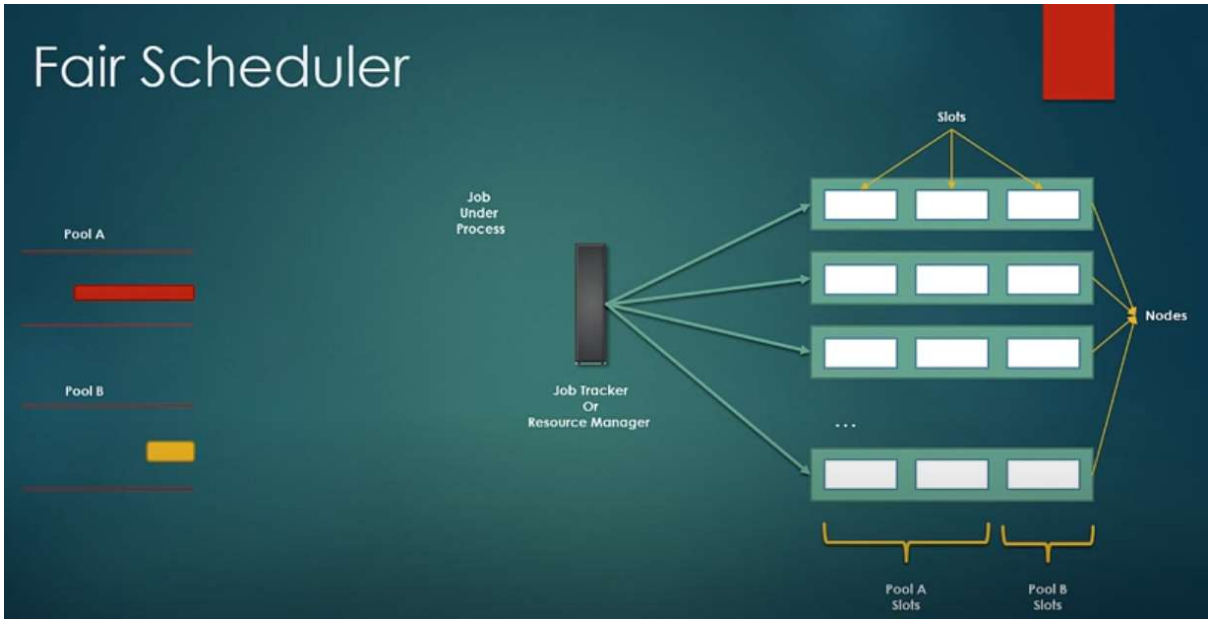
Fair Scheduler takes Scheduling decisions on the basis of memory, we can configure it to work with CPU also.

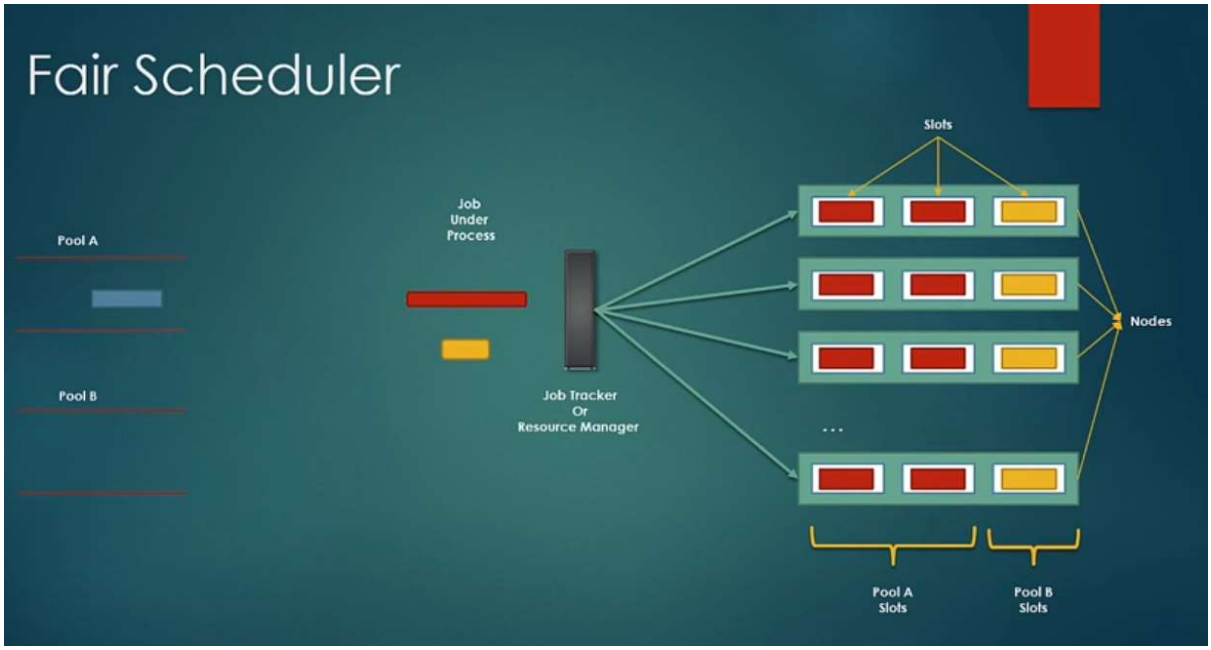
It is similar to Capacity Scheduler but the major thing is that in Fair Scheduler whenever any high priority job arises in the same queue, the task is processed in parallel by replacing some portion from the already dedicated slots.

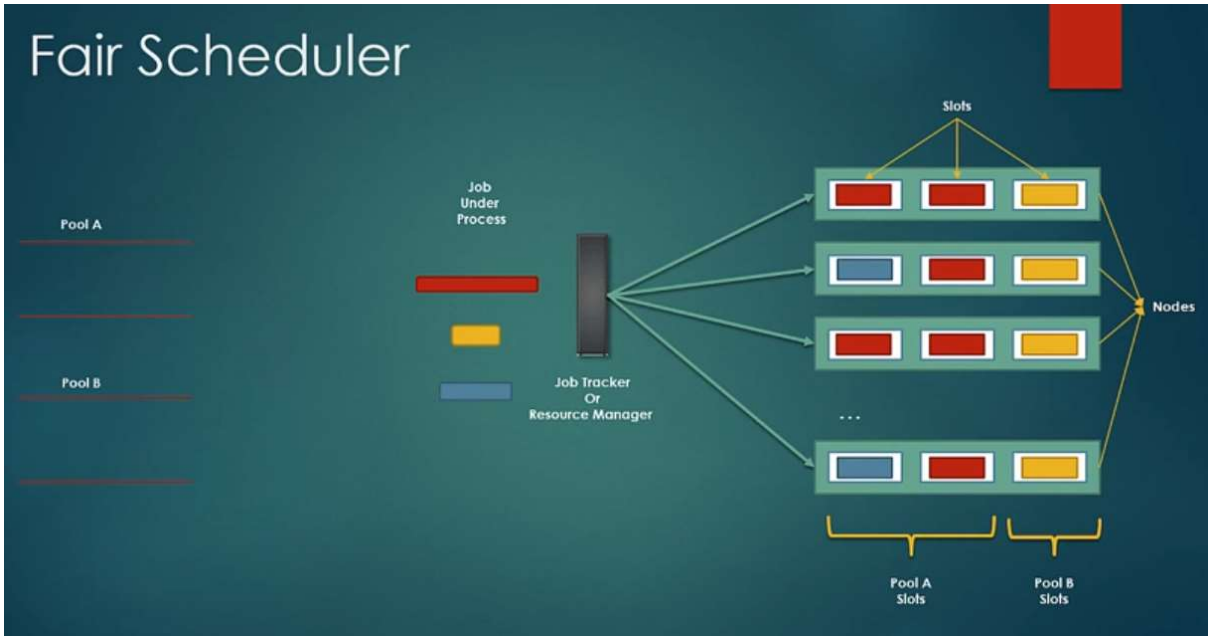



Capacity Scheduler
Slots
Job
Under
Process
Organisation A
Organisation B
Job Tracker
Or
Resource Manager
Org. A
Slots
Org. B
Slots


Capacity Scheduler
Slots
Job
Under
Process
Organisation A
Organisation B
Job Tracker
Or
Resource Manager
Nodes
Customisable Features:
1. Capacity Guarantees.
2. Elasticity.
3. Security.
Org. A
Slots
Org. B
Slots


Fair Scheduler
Slots
Job
Under
Process
Pool A
Pool B
Job Tracker
Or
Resource Manager
Nodes
...
Pool A
Slots
Pool B
Slots


Fair Scheduler
Slots
Job
Under
Process
Pool A
Pool B
Job Tracker
Or
Resource Manager
Nodes
...
Pool A
Slots
Pool B
Slots



4)What is HIVE? And Explain.

Apache Hive is a data warehouse and an ETL tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS).
HIVE is a interface between HDFS and User.

Components of Hive:

1. **HCatalog –**
   It is a Hive component and is a table as well as a store management layer for Hadoop. It enables user along with various data processing tools like Pig and MapReduce which enables to read and write on the grid easily.
2. **WebHCat –**
   It provides a service which can be utilized by the user to run Hadoop MapReduce, Pig, Hive tasks or function Hive metadata operations with an HTTP interface.

Modes of Hive:

1. **Local Mode:**
   It is used, when the Hadoop is built under pseudo mode which has only one data node.
2. **Map Reduce Mode:**
   It is used, when Hadoop is built with multiple data nodes and data is divided across various nodes.

5)What is PIG? Write down execution modes of PIG.

Pig is a high-level platform or tool which is used to process the large datasets
It provides a high-level scripting language, known as *Pig Latin Language.*
Pig Engine has two type of execution environment i.e. a *local execution environment* in a single JVM and *distributed execution environment* in a Hadoop Cluster.

Features of Apache Pig:

- Easy to learn, read and write. Especially for SQL-programmer, Apache Pig is a boon.
- Apache Pig is extensible so that you can make your own user-defined functions and process.
- Fewer lines of code.
- Pig can handle the analysis of both structured and unstructured data.

Types of Data Models in Apache Pig:

It consist of the 4 types of data models as follows:

- **Atom**: It is a atomic data value which is used to store as a string.
- **Tuple**: It is an ordered set of the fields.
- **Bag**: It is a collection of the tuples.
- **Map**: It is a set of key/value pairs.

Modes:

1.Local Mode:

- In this mode of execution, we need a single machine and all files are installed and run using your localhost and file system.
- This mode is used for testing and development purposes. The local mode does not need HDFS or Hadoop.

3. Mapreduce Mode:

- Mapreduce is the default mode of the Apache Pig Grunt shell.
- In this mode, we need to load data in HDFS and then we can perform the operation.

Apache Pig Execution Methods

A user can execute Apache Pig Latin scripts in three ways as mentioned below.

- 1. Interactive Mode (Grunt shell)

  In this mode, a user can interactively run Apache Pig using the Grunt shell. To invoke Grunt Shell, run Pig command. Users can submit commands and get a result there only.

- Batch Mode (Script): In this mode, a user can run Apache Pig in batch mode by creating a Pig Latin script file and running it from local or MapReduce mode.
- Embedded Mode: We can define our own functions called as UDF(Usser Defined Functions).

Grunt

Grunt shell is a shell command.The Grunts shell of Apace pig is mainly used to write pig Latin scripts. Pig script can be executed with grunt shell which is native shell provided by Apache pig to execute pig queries.

We can invoke shell commands using **sh** and **fs.**

Syntax of **sh** command:

```
grunt> sh ls
```

, User Defined Functions

Apache Pig - User Defined Functions (tutorialspoint.com)

[redacted]

apache pig provides high level scripting language known as apache pig latin language.
we need to transform Pig Latin statements into MapReduce jobs.
Data Model in Pig Latin:
A bag, what we call a collection of tuples.
A tuple, what we call an ordered set of fields.
A field, what we call a piece of data.

Pig Latin Datatypes
int, long, float, double boolean,bytearray,etc.

arithmetic and comparision operators.

The Apache Pig Operators is a high-level procedural language for querying large data sets using Hadoop and the Map-Reduce Platform.

A Pig Latin statement is an operator that takes a relation as input and produces another relation as output.

These operators are the main tools for Pig Latin provides to operate on the data.

They allow you to transform it by sorting, grouping, joining, projecting, and filtering.

The Apache Pig operators can be classified as :

Relational Operators :

Relational operators are the main tools Pig Latin provides to operate on the data.

Some of the Relational Operators are :

LOAD: The LOAD operator is used to loading data from the file system or HDFS storage into a Pig relation.

FOREACH: This operator generates data transformations based on columns of data. It is used to add or remove fields from a relation.

FILTER: This operator selects tuples from a relation based on a condition.

JOIN: JOIN operator is used to performing an inner,

JOIN: JOIN operator is used to performing an inner, equijoin join of two or more relations based on common field values

ORDER BY: Order By is used to sort a relation based on one or more fields in either ascending or descending order using ASC and DESC keywords.

GROUP: The GROUP operator groups together the tuples with the same group key (key field).

COGROUP: COGROUP is the same as the GROUP operator. For readability, programmers usually use GROUP when only one relation is involved and COGROUP when multiple relations are reinvolved.

Diagnostic Operator :

The load statement will simply load the data into the specified relation in Apache Pig.

To verify the execution of the Load statement, you have to use the Diagnostic Operators.

Some Diagnostic Operators are :

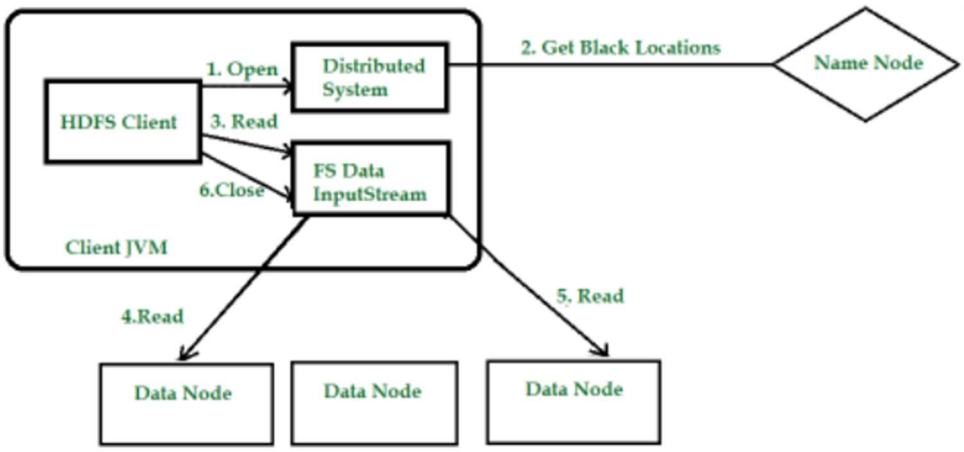
DUMP: The DUMP operator is used to run Pig Latin statements and display the results on the screen.

DESCRIBE: Use the DESCRIBE operator to review the schema of a particular relation. The DESCRIBE operator is best used for debugging a script.

ILLUSTRATE: ILLUSTRATE operator is used to review how data is transformed through a sequence of Pig Latin statements. ILLUSTRATE command is your best friend when it comes to debugging a script.

Anatomy of File Write and Read,

Anatomy of File Read in HDFS



Step 1: The client opens the file it wishes to read by calling open() on the File System Object.

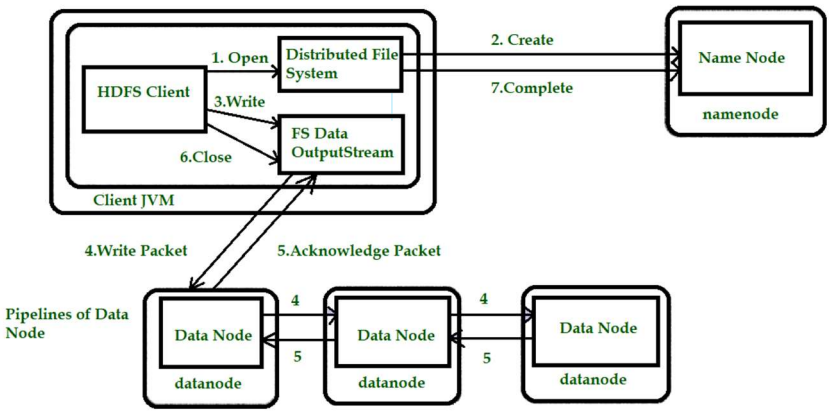
Step 2: Distributed File System(DFS) calls the name node, using remote procedure calls (RPCs), to determine the locations of the first few blocks in the file. For each block, the name node returns the addresses of the data nodes that have a copy of that block.
The DFS returns an FSDataInputStream to the client for it to read data from.
FSDataInputStream in turn wraps a DFSInputStream, which manages the data node and name node I/O.
Step 3: The client then calls read() on the stream. DFSInputStream, which has stored the info node addresses for the primary few blocks within the file, then connects to the primary (closest) data node for the primary block in the file.
Step 4: Data is streamed from the data node back to the client, which calls read() repeatedly on the stream.
Step 5: When the end of the block is reached, DFSInputStream will close the connection to the data node, then finds the best data node for the next block. **Step 6:** When the client has finished reading the file, a function is called, close() on the FSDataInputStream.

Anatomy of File Write in HDFS

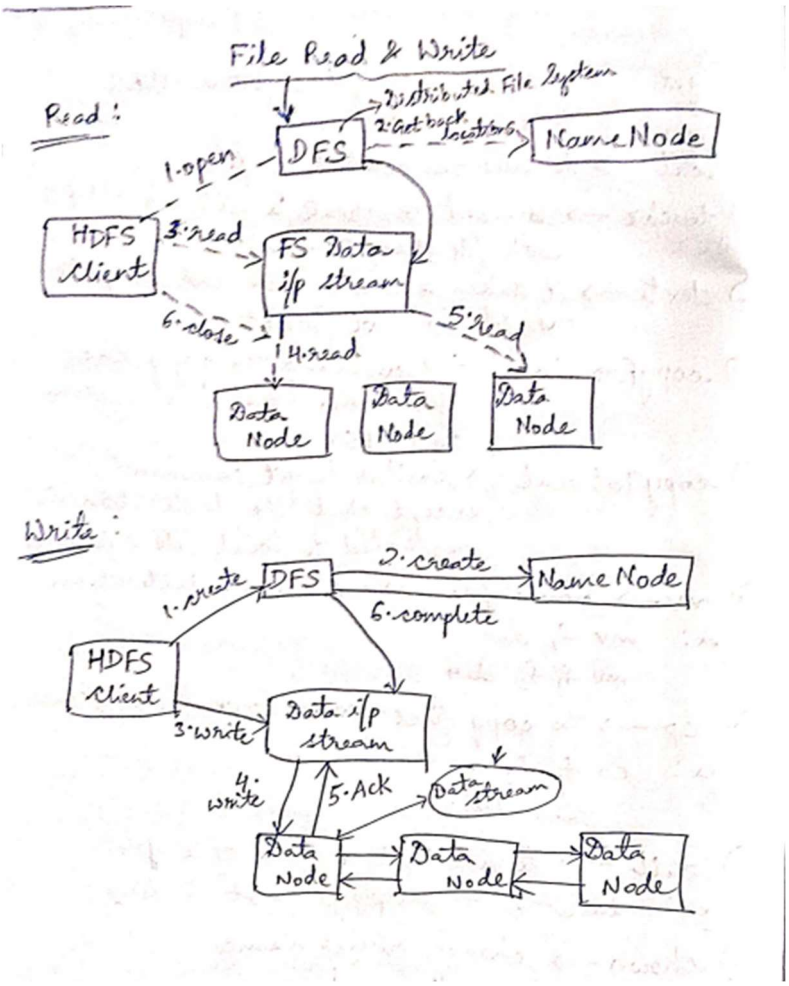
HDFS follows the Write once Read many times model. In HDFS we cannot edit the files which are already stored in HDFS, but we can append data by reopening the files.

Step 1: The client creates the file by calling create() on DistributedFileSystem(DFS).

Step 2: DFS makes an RPC call to the name node to create a new file in the file system's namespace. The name node performs various checks to make sure the file doesn't already exist, if these checks pass, the name node prepares a record of the new file; otherwise, the file can't be created and therefore the client is thrown an error i.e. IOException.
The DFS returns an FSDataOutputStream for the client to start out writing data to.



Step 3: Because the client writes data, the DFSOutputStream splits it into packets, which it writes to an indoor queue called the info queue. The data queue is consumed by the DataStreamer, which is liable for asking the name node to allocate new blocks by picking an inventory of suitable data nodes to store the replicas. The list of data nodes forms a pipeline, and here we'll assume the replication level is three, so there are three nodes in the pipeline. The DataStreamer streams the packets to the primary data node within the pipeline, which stores each packet and forwards it to the second data node within the pipeline.
Step 4: Similarly, the second data node stores the packet and forwards it to the third (and last) data node in the pipeline.
Step 5: The DFSOutputStream sustains an internal queue of packets that are waiting to be acknowledged by data nodes, called an "ack queue".
Step 6: This action sends up all the remaining packets to the data node pipeline and waits for acknowledgments before connecting to the name node to signal whether the file is complete or not.



Comparison of hive and pig with traditional databases

CRITERIA	HIVE	PIG	SQL
Languages used	Uses HiveQL, a declarative language	Uses Pig latin, a procedural data flow languages	SQL itself is a declarative language
Definition	An open source built with an analytical focus used for Analytical queries	An open source and high-level data flow language with a Multi-query approach	General purpose database language for analytical and transactional queries
Developed by	Facebook	Yahoo	Oracle
Suitable for	Batch processing OLAP (Online Analytical Processing)	Complex & nested data structure	Business demands for fast data analysis
Operational for	Structured data	Structured and semi-structured data	Relational database management
Compatibility with MapReduce	Yes	Yes	Yes
Schema Support	Support schema for data insertion	Doesn't support schema	Strictly support schema for data storage
Mainly Used by	Data Analysts	Researchers and Programmers	Data Analysts, Data Scientists, and Programmers

Hive Services

The following are the services provided by Hive:-

- Hive CLI - The Hive CLI (Command Line Interface) is a shell where we can execute Hive queries and commands.
- Hive Web User Interface - The Hive Web UI is just an alternative of Hive CLI. It provides a web-based GUI for executing Hive queries and commands.
- Hive MetaStore - It is a central repository that stores all the structure information of various tables and partitions in the warehouse. It also includes metadata of column and its type information, the serializers and deserializers which is used to read and write data and the corresponding HDFS files where the data is stored.
- Hive Server - It is referred to as Apache Thrift Server. It accepts the request from different clients and provides it to Hive Driver.
- Hive Driver - It receives queries from different sources like web UI, CLI, Thrift, and JDBC/ODBC driver. It transfers the queries to the compiler.
- Hive Compiler - The purpose of the compiler is to parse the query and perform semantic analysis on the different query blocks and expressions. It converts HiveQL statements into MapReduce jobs.
- Hive Execution Engine - Optimizer generates the logical plan in the form of DAG of map-reduce tasks and HDFS tasks. In the end, the execution engine executes the incoming tasks in the order of their dependencies.

What is the Hive Metastore?

The Hive metastore is simply a relational database. It stores metadata related to the tables/schemas you create to easily query big data stored in HDFS. When you create a new Hive table, the information related to the schema (column names, data types) is stored in the Hive metastore relational database. Other information like input/output formats, partitions, HDFS locations are all stored in the metastore.

How does the Hive Metastore work?

The Hive metastore acts as a central repo for Hive metadata. The Hive driver connects to the metastore and queries information related to the schema you create so that you can run SQL like queries (HQL) against the underlying Hadoop cluster. Remember that Hive simply sits on top of HDFS, it is not an actual form of storage for the big data you store via HDFS, HBase, etc.

By default, Hive uses Derby SQL server for the metastore database. This runs as a single process and is not recommended for production. Using other supported relational engines like MySQL or Postgres is necessary for running Hive on a production cluster.

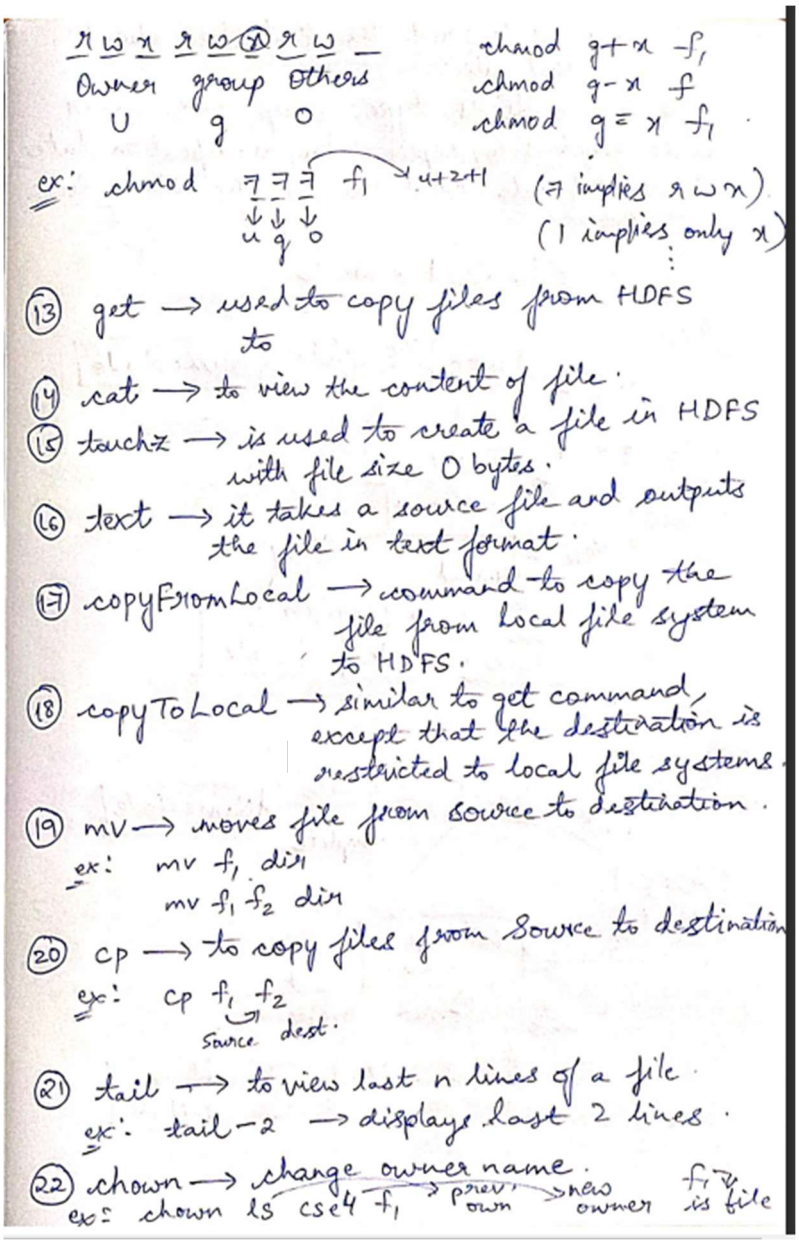
HDFS shell commands

HDFS Commands :-

1. Version check → version details
2. List command → ls / dir
   - ↳ list of files
   - of directories in current directory.
3. df command → disk free space info.
   (displays free space at given HDFS destination)
4. count → count the no. of directories, files and bytes under the paths that match the specified file pattern.
5. fsck → to check the health of the Hadoop file system.
6. balancer → run a cluster balancing utility.
7. mkdir → to create directory.
8. put → copy files from single/multiple sources from local system to destination file system.
9. du → disk usage.
   (displays size of the files & directories contained in the given directory).
10. rm → to remove file from HDFS (DOS/Linux)
11. rm -r → to remove entire directory and all its contents from HDFS.
12. chmod → change mode of operation

    add + r w x ← set =

read write execute no permission
4 2 1 0



(13) get → used to copy files from HDFS to

(14) cat → to view the content of file.

(15) touchz → is used to create a file in HDFS with file size 0 bytes.

(16) text → it takes a source file and outputs the file in text format.

(17) copyFromLocal → command to copy the file from local file system to HDFS.

(18) copyToLocal → similar to get command, except that the destination is restricted to local file systems.

(19) mv → moves file from source to destination.
ex: mv f1 dir
mv f1 f2 dir

(20) cp → to copy files from source to destination
ex: cp f1 f2
source dest.

(21) tail → to view last n lines of a file.
ex: tail -2 → displays last 2 lines.

(22) chown → change owner name.
ex: chown ls cse4 f1 → prev own → new owner; f1 → file