RAISONI GROUP
—a vision beyond—

Title :- To install Hadoop framework, configure it and setup a single node cluster. Use web based tools to monitor your Hadoop setup.

Theory :-

- Stand Alone mode :- Hadoop is a distributed software and is designed to run on a commodity of machine. However, we can install it on single node in stand-alone mode.

- Pseudo Distributed Mode :- In this mode also Hadoop software is installed on a Sigle Node Various deamons of Hadoop will run on the same machine as seperate java processes.

- Fully Distributed Mode :- In Fully Distributed Mode, the daemons Name Node, Job Tracker, Secondary Name Node. All the daemons that are NameNode Datanode, Secondary Name node, Resource Manager, Node-Manager etc.

Hadoop Installation :- Ubunta Operting System in stand-alone mode

1) sudo apt-get update

2) In this step, we will install

3) The Oracle JDK is the official JDK however it is no longer provided by Oracle as a default installation for Ubuntu.

4) Now, let's setup a new user account for Hadoop

5) Download the latest Hadoop distribution.

6) Untar the file :-

```
tar xvzf hadoop-2.7.0.tar.gz
```

7) Rename the folder to hadoop2

```
mv hadoop-2.7.0 hadoop2
```

8) Edit configuration file

9) This finishes the hadoop setup in stand-alone mode

10) Let us run a sample hadoop program that is provided to you in the download package.

RAISONI GROUP
a vision beyond

Hadoop Installation : Pseudo Distributed Mode

Choosing a mirror site : When you visit the Apache Hadoop download page, you will see a list of mirror sites that you can choose from. These mirror sites are maintained by organizations around the world that host copies of the hadoop distribution for faster and more reliable downloads.

Copying the download link : Once you have chosen a mirror site, you will see a list of files that you can download. Look for the file that matches your operating system and architecture. To copy the download link for the file, right click on the link and select "copy link address" or similar option depending on your browser. This will copy the download link to your clipboard, which you can use to download the file using a command-line tool such as wget.

Downloading using wget : If you prefer to download the hadoop distribution using wget from the command prompt, you can open a terminal or command prompt and type command

RAISONI GROUP
—a vision beyond—

2) Edit the file /home/Hadoop-dev/Hadoop2/etc/hadoop/hdfs-side.xml as below

3) We need to setup password less login so that the master will be able to do passord-less ssh to start the deamons on all the slaves.

4) We can run Hadoop jobs locally or on YARN in this mode. In this Post, we will focus on running the jobs locally

5) Format the file system, when we formal namenode it format the meta-deta related to data-node By doing that all the information on the datamode are lost and they can be reused for newdata

6) Start the daemons

7) You can check whether the deamons are runnigor not by issuing Jps command

8) This finsihes the installation of Hadoop in pseudodistributed mode

9) Let us run the same example we can in the previous blog post

RAISONI GROUP
—a vision beyond—

10) Stop the daemons when you are done executing the jobs with below command

sbin/stop-dfs.sh.

RAISONI GROUP
— a vision beyond —

Title :- To implement file management tasks in hadoop HDFS like adding retriving and deleting files

Theory :-

The entire MapReduce program can be fundamentally divided into three parts

- Mapper Phase Code
- Reducer Phase Code
- Driver Code.

1) Mapper Phase Code:

The mapper phase is the first phase of the MapReduce program. In this phase, the input data is read and processed in parallel across multiple nodes in the hadoop cluster.

2) Reducer Phase Code:

The reducer phase is the scond phase of the Map Reduce program. In this phase the output from the mapper phase is collected and processed to produce the final output.

The reducer code is responsible for taking the key-value pairs generated by the mapper and reducing them to smaller set of key-value pairs.

RAISONI GROUP
—a vision beyond—

3) Driver Code

The driver code is the code that controls the execution of the MapReduce program. It is responsible for setting us the configuration parameter, input / output paths and the mapper and reducer classes.

The driver code also define the number of reducer, input format and output formats. The driver code typically runs on the client machine and it submits the MapReduce job to the Hadoop cluster for execution.

- Mapper Code

```
public static class Map extends Mapper < long Writable,
  Text, IntWritable > {
public void map (Long Writable key, Text value,
  Context context ) throws
IOException, Interrupted Exception {
  String line = value.to String ();
  String Tokenizer tokenizer = new String Tokenizer (line)
  while (tokenizer.has More Token, ()) {
    value.set (tokenizer.next Token ());
    context.write (value, new Int Writable (1));
}
```

RAISONI GROUP
—a vision beyond—

- Input
  - The key is nothing but the offset of each line in the text file : LongWritable
  - The value is each individual line : Text
- Output
  - The key is the tokenized words : Text
  - We have the hardcoded value in our case which is 1 : IntWritable
- Reducer Code :

```
public static class Reduce extend Reducer <Text,
IntWritable , Text, IntWritable > {
    public void reduce (Text key, Iterable <IntWritable>
      values , Context context)
throws IOException , InterruptedException {
    int sum = 0;
    for ( IntWritable x : values)
    {
       sum += x.get ();
    }
    context .write (key, new IntWritable (sum));
    }
    }
```

RAISONI GROUP
— a vision beyond —

- Input
  - The key nothing but those unique words which have been generated after the sorting and shuffling phase : Text
  - The value is a list of integer corresponding to each key : IntWritable.
  - Example - Bear, [1,1] etc.
- Output
  - The key is all the unique words present in the input text file : Text
  - The value is the number of occurrence of each of the unique words : IntWritable
  - Example - Bear, 2;

- We have aggregated the values present in each of the list corresponding to each key and produced the final answer.

- In general, a sigle reducer is created for each of the unique words, but, you can specify the number of reducer in mapred-site.xml.

In Hadoop MapReduce, both the input and output of the reducer phase is a key-value pair. The key value-pair generated by the mapper phase for further processing.

RAISONI GROUP
— a vision beyond —

- Input
  - The key nothing but those unique words which have been generated after the sorting and shuffing phase : Text
  - The value is a list of integer corresponding do each key : Int Writable.
  - Example - Bear , [1,] etc.
- Output
  - The key is all the unique words present in the imput text file : Text
  - The value is the number of occurrence of each of the unique words : Int Writable
  - Example - Bear , 2)

- We have aggregated the values present in each of the list corresponding to each key and produced the final answer.

- In general , a sigle reducer is created for each of the unique words , but , you can specify the number of reducer in mapred - site . xml.

In Hadoop MapReduce , both the input and output of the reducer phase is a key - value pair. The key value - pair generated by the mapper phase for further processing.

RAISONI GROUP
—a vision beyond—

The reducer function then performs its computation on the input key-value pairs and generated a set of output key-value pairs. The output key-value pairs generated by the reducer function are written to the output file or HDFS. The output key-value pair can be of different type, depending on the requirement of the application.

- In the driver class, we set the configuration of our MapReduce job to run in Hadoop
- We specify the name of the job, the data type of input/output of the mapper and reducer
- The path of the input and output folder is also specified
- The method setInputFormatClass() is used for specifying that how a Mapper will read the input data or what will be the unit of work
- Here, we have chosen TextInputFormat so that single line is read by the mapper at a time from the input text file.
- The main() method is the entry point for the driver. In this, we instantiate a new Configuration object for the job.

RAISONI GROUP
—a vision beyond—

Title :- To implement a word count application using the MapReduce API

Theory :-

Weather Report POC - Map Reduce Program to analyse time - temperature statistics. and generate report with max/min temperature

Problem statement

1. The system receives temperature of various cities of USA captured at regular interval of time on each day in input file.

2 System will process the input data file and generates a report with Maximum and Minimum temperature of each day along with time.

3. Generates a seperate output report for each city

Austin - r- 00000
Boston - r - 00000
Newjersy - r- 00000
Baltimore - r- 00000
California - r- 00000
Newyork - r- 00000

RAISONI GROUP
— a vision beyond —

Expected output:
In each output file record should be like this 25-Jan-2014 Time: 12:34:542 MinTemp=22.3 Time 05:12:345

Schema of record set: CA-25-Jan-2014 00:12:345 15.7 01:19:345 23.1

Mapper class and map method
The very first thing which is required for any map reduce problem is to understand what will be the type of keyIn, valueIn key out for the given Mapper class and followed by type of map method parameter

- public class Whether Forcast Mapper extend Mapper <Object, Text, Text, Text)
- Object (key In) - offset for each line, line number 1, 2 ....
- Text (valueIn) - Whole string for each line
- Text (key Out) - City information with date information as string
- Text (value out) - Temperature and time information which need to be passed to reducer as string
- con is context where we write mapper output and it is used by reducer.

RAISONI GROUP
—a vision beyond—

Reducer class and reducer method :-

- public class WhetherForcastReducer extend, Reducer < Text, Text, Text, Text)
- Text (KeyIn) - it is same as value Out of Mapper
- Text (ValueIn) - it is same as value Out of Mapper
- Text (KeyOut) - date as string
- Text (ValueOut) :- reducer write max and min temperature with time as string
- public void reduce (Text key, Iterable < Text > value, (Context context)
- Text key is value of mapper output
- Iterable < Text) value - value store multiple temperature value for a given city and date.
- context object is where reducer write it's processed outcome and finally written in file.

Multiple Outputs :-

In general, reducer generates output file however in this use case we want to generate multiple output files. In order to deal with such scenerio we need to use Multiple Outputs which provides a way to write multiple file depending on reducer out come.

RAISONI GROUP
—a vision beyond—

-Lets create a Map/Reduce project in eclipse and create class file name it as Calculate Max And Min Temperature With Time. For simplicity, here written map-per and reducer class as inner static class.

RAISONI GROUP
—a vision beyond—

Title :- Creating the HDFS table and loading them in Hive

Theory :-

Creating the HDFS table and loading them in Hive.

HIVE table provide us the schema to store data in various formats (like CSV). Hive provides multiple way to add data to the table. We can use DML queries in Hive to import or add data to the table. One can also directly put the table into the hive with HDFS command. In case we have data in Relational Database like MySQL, ORACLE To perform the below operation make sure your hive is running

Step 1 : Start all your Hadoop Daemon.

Step 2 : Launch hive from terminal.

In hive with DML statement, we can add to the Hive table in 2 different ways.

- Using INSERT Command
- Load Data statement.

RAISONI GROUP
—a vision beyond—

1) Using INSERT Command

Syntax:

Insert into table <table-name> value ;

Command:

```
CREATE TABLE IF NOT EXISTS student (
Student-Name STRING,
Student-Rollno INT,
Student-Marks FLOAT)
```

INSERT Query :-

```
INSERT INTO TABLE student VALUES ('Dikshant',1,'91'),
('Akshat',2,'96'), ('Dhruv',3,'90');
```

2) Load Data statement

Hive provide us the functionality to load precreated table entries either from our local file system or from HDFS -

Syntax :-

```
LOAD DATA [LOCAL] INPATH <The table data location>
IN TO TABLE <table-name>;
```

RAISONI GROUP
—a vision beyond—

Note :-

- The LOCAL switch specifies that the data we are loading is available in our Local File system
- The OVERWRITE switch allow us to overwrite the table data

Command

cd / home / dikshant / Document

touch data . csv

nano data . csv

RAISONI GROUP
— a vision beyond —

Title :- To create HDFS tables and load them in Hive and Implement joining of tabale in Hive.

Requirement :-

You have two table named as A and B and you want to perform all type of join in hive. It will help you to how join work in hive.

Theory :-

Step 1 :- Input files

Create an HDFS directory where you will store the data for your table.

Create table in Hive using the CREATE TABLE command. You can specify the table name, column name, and data type.

Step 2 : Loading into files into Hive.

hadoop fs - mkdir bdps/sample_files

hadoop fs - mkdir bdps/sample_files/A

hadoop fs - mkdir bdps/sample_file/B

hadoop fs - put A/A.txt bdps/sample_files/A/

hadoop fs - put B/B.txt bdps/sample_files/B/

RAISONI GROUP
— a vision beyond —

Now let's create two hive table A and B for both the files, using below commands.

```
CREATE SCHEMA IF NOT EXISTS bdp;
CREATE EXTERNAL TABLE IF NOT EXISTS bdp.A
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION

CREATE EXTERNAL TABLE IF NOT EXISTS bdp.B
ROW FORMAT DELIMITED
FIELDS TERMINATED
STORED AS TEXTFILE
```

A. Inner Join :-

Sometimes it is required to have only common records out of two datasets. Now we have two table A & B we are joining based on key which is id

```
select * from
(select * from bdp.A) T1
JOIN
(select * from bdp.B) T2
ON T1.id = T2.id;
```

RAISONI GROUP
—a vision beyond—

B. Left Join

this type of join is performed when we want to look up something from other dataset, the best example would be fetching a phone no of an employee from other dataset based on employee code.

```
select * from
(select * from bdp.A) T1
LEFT JOIN
(select * from bdp.B) T2
ON T1.id = T2.id;
```

C. Right Join

This type of join is preformed when we want to get the data of look up table with only matching records of left table.

```
select * from
(select * from bdp.A) T1
RIGHT JOIN
(select * from bdp.B) T2
ON T1.id = T2.id;
```

RAISONI GROUP
—a vision beyond—

D. Full Join

When it is needed to get all the matched and unmatched records of two datasets, we can use full join. All data from left as well as from right dataset will appear in result set. Non-matching records will have null have in respective column

```
select * from
(select * from bdp.A)T1
FULL JOIN
(select * from bdp.B) T2
ON T1.id = T2.idj
```

RAISONI GROUP
—a vision beyond—

Title :- To install, deploy & configure Apache Spark cluster. To select the field from the dataset using spark SQL. To explore Spark shell and read from HDFS.

Requirement :- Ubuntu 16.04 or higher installed on virtual machine.

Theory :-

What is Apache Spark?

It is open source and distributed processing system used for big data workloads. Spark is fast, general engine and powerful engine for big data processing. Apache Spark has master worker architecture and a cluster manager.

- Master Daemon
- Worker Daemon
- Cluster Manager

An Apache Spark cluster used to have a master and one or more than one Worker / slaves.

Steps for istallation of Apache Spark 3.1.1 cluster on Hadoop 3.2

RAISONI GROUP
—a vision beyond—

step 1

Create two clone of the Oracle VM virtual Box Machine that has been earlier created. Select option "Generate new MAC addresses for all network adapter" in MAC address Policy. And also choose the option Full Clone in clone type

Step 2

Go to the setting option of virtual machine and make the following network configuration on Adapter 2.

step 3:

You need to set the hostname of each virtual machine. Open the etc/hostname file and type the name of the machine in it and save. Run the following command one each virtual machine

$ sudo nano/etc/hostname.

Step 4

To figure out IP address of the virtual machine run the following command

$ ip addr

RAISONI GROUP
a vision beyond

step 5

In this step, I edit the hosts file and added IP address and hostname information saved it and reboot the machine. Run the following command on all the machine

$ sudo nano /etc/hosts

Step 6

Run the following command on all the machine

$ sudo apt-get update $

$ sudo apt install openjdk-8-jdk

$ java - version

step 7:

Intall Scala on all the machines

Run the following command

$ sudo get-apt install scala

To check the version of Scala. run the following command

$ scala - version

RAISONI GROUP
—a vision beyond—

Step 8

Now configure Open SSH server-client on master. To configure the Open SSH server-client run the following command

$ sudo apt-get install openssh-server openssh-client

The next step is to generate key pair

$ ssh-keygen -t rsa -P

Run the following command to authorize the key

$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorised_key

Step 9

Download the stable version of Apache Spark. I will install spark-3.1.1 with Hadoop 3.2 To download spark 3.1.1 with Hadoop 3.2

Now run the following command to untar the spark tar file

$ tar xvf spark-3.1.1-bin-hadoop3.2.tgz

Add the following line to the file and save

export PATH = $PATH :/usr/local/spark/bin.

RAISONI GROUP
— a vision beyond —

Step 10

To step Apache spark Master configuration edit spark-env.sh file. Traverse to the spark conf folder and make a copy of the spark env.sh template file as a spark-ens.sh

$ cd /usr/ local / spark / conf $ cp spark-env.sh

Now, to edit the spark-env.sh configuration file run the following command

$ sudo -nano spark-env.sh

Add the following parameter at the end of the file save and exit

export SPARK_Master_HOST = <Master-IP>

Step 11

Now our Apache Spark cluster is ready to start the apache spark cluster the run the following command

$ cd /usr/ local / spark $ ./ sbin / start-all.sh

RAISONI GROUP
—a vision beyond—

To explore Spark shell and read from HDFS

1. Step 1 :- Import the module.
   Step 2 :- Create Spark session
   Step 3 :- Create schema
   Step 4 :- Read CSV File from HDFS
   Step 5 :- To view the schema.

RAISONI GROUP
—a vision beyond—

Title :- To install and run Pig and then write Pig Latin scripts to sort, group join, project and filter your data.

Theory :-

What is Apache Pig

Apache Pig is an abstraction over Map Reduce It is tool / perform which is used to analyze larger sets of data representing them as flows. Pig is generally used with Hadoop To write data analysis program, Pig provide a high-level language known Pig Latin This language provide various operator using which programmer can develop their own functions for reading writing and processing data

To analyze data using Apache Pig, programmer need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig have a component known as Pig Engine that accept the Pig Latin scripts as input and converts these scripts in Map Reduce jobs.

RAISONI GROUP
—a vision beyond—

Pig Feature :-

- Rich set of operator :- It provide many operator to perform operation like join, sort, filter etc
- Ease of programming - Pig Latin is similar to SQL and it is easy to write a Pig script if you are good at SQL
- Extensibility - Using the existing operator, user can develop their own function to read process and write data
- UDF - Pig provide the facility to create User-defined Function in other programming language
- Handle all kinds of data - Apache pig analyzes all kind of data, both structured as well as unstructed

To install and run Pig and write Pig latin scripts you will need to flow several step

Install java :- Pig require Java to run, so you will need to install Java on your computer if you haven't already

RAISONI GROUP
—a vision beyond—

Download Pig : Next you will need to download Pig from the Apache website

Start Pig :- Once you have set up environment variable, you can start Pig by running the Pig

Relations :- In Pig data is organised into relation which are similar to table in a database

Loading and storing data :- You can load data into Pig from a variety of source, including Hadoop Distributed File System (HDFS) local file and database

Filtering and projecting data : You can filter data in relation by specifying a condition that m