EXPERIMENT -01

AIM: Write the steps for installation of Hadoop for windows.

Step1:

- Prerequisites before installing JDK11.

JDK: https://www.oracle.com/java/technologies/javase-downloads.html

Step2:

- Setting environment to the device.
- **Set JAVA_HOME Environment Variable:**
    - Open the Start menu, search for "Environment Variables", and select "Edit the system environment variables".
    - In the System Properties window, click on the "Environment Variables" button.
    - Under System variables, click "New" to add a new variable.
        - Variable name: JAVA_HOME
        - Variable value: The path to your JDK installation directory (e.g., C:\Program Files\Java\jdk-<version>).
    - Click OK.
- **Update the PATH Variable:**
    - In the System Variables section, find the Path variable, select it, and click "Edit".
    - Click "New" and add the path to the bin directory of your JDK installation (e.g., C:\Program Files`\Java\jdk-<version>\bin`).
    - Click OK to close all dialogs.
- **Verify the Setup:**
    - Open a Command Prompt and type `java -version` and `javac -version` to verify that the installation was successful and that the JDK is correctly set up.

Step3: install 7-zip

https://7-zip.org/download.html

- **After that Hadoop set-up.**

Step 4:

Hadoop is a well-known big data processing system for storing and analysing enormous volumes of data. It's an open-source project that you can use for free. If you're new to Hadoop, you may find the installation process difficult.

Download Hadoop

Hadoop can be downloaded from the Apache Hadoop website. Make sure to have the latest stable release of Hadoop. Once downloaded, extract the contents to a convenient location.

Hadoop:

https://hadoop.apache.org/releases.html Or

https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.2.2/hadoop-3.2.2-aarch64.tar.gz

After unzip the hadoop-3.2.2.tar file using 7-zip tool from

downloads. After that unzip for hadoop-3.2.2 one more tar file in

download.

And cut the hadoop-3.2.2 folder and placed in C drive.

Step 5: Set Environment Variables

You must configure environment variables after downloading and unpacking Hadoop. Launch the Start menu, type "Edit the system environment variables," and select the result. This will launch the System Properties dialogue box. Click on "Environment Variables" button to open.
Click "New" under System Variables to add a new variable. Enter the variable name "HADOOP_HOME" and the path to the Hadoop folder as the variable value. Then press "OK."
Then, under System Variables, locate the "Path" variable and click "Edit." Click "New" in the Edit Environment Variable window and enter "%HADOOP_HOME%bin" as the variable value. To close all the windows, use the "OK" button.

Step6: Setup Hadoop

You must configure Hadoop in this phase by modifying several configuration files.

Navigate to the "etc/hadoop" folder in the Hadoop folder. You must make changes

to three files:

- **core-site.xml**

```
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
```

-

- **hdfs-site.xml**

```
<property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>/D:/bigdata/hadoop-3.2.2/dfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>/D:/bigdata/hadoop-3.2.2/dfs/datanode</value>
  </property>
```

- **mapred-site.xml**

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9870</value>
  </property>
</configuration>
```

-

Open each file in a text editor and edit the following properties:

In mapred-site.xml

Save the changes in each file.

Step 7: Format Hadoop NameNode

You must format the NameNode before you can start Hadoop. Navigate to the Hadoop bin folder using a command prompt. Execute this command:

```
hadoop namenode -format
```

Step 8: Start Hadoop

To start Hadoop, open a command prompt and navigate to the Hadoop bin folder. Run the following command:

```
start-all.cmd
```

This command will start all the required Hadoop services, including the NameNode, DataNode, and JobTracker. Wait for a few minutes until all the services are started.

Step 9: Verify Hadoop Installation

To ensure that Hadoop is properly installed, open a web browser and go to http://localhost: HYPERLINK "http://localhost:50070/" HYPERLINK "http://localhost:50070/"9870 HYPERLINK "http://localhost:50070/" HYPERLINK "http://localhost:50070/"/. This will launch the web interface for the Hadoop NameNode. You should see a page with Hadoop cluster information.

EXPERIMENT -02

AIM: Write syntax with an example of commands for Hadoop File System. (Hadoop Commands).

- Print the Hadoop version:

Syntax: hadoop -version

Output:

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

E:\hadoop-3.3.6\etc\hadoop>hadoop -version
java version "11.0.24" 2024-07-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.24+7-LTS-271)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.24+7-LTS-271, mixed mode)

E:\hadoop-3.3.6\etc\hadoop>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

E:\hadoop-3.3.6\etc\hadoop>
```

Now open https://localhost:9870 in chrome

2) To create a directory :

Syntax: hadoop fs -mkdir /<name of dir>

```
command [genericOptions] [commandOptions]

E:\hadoop-3.3.6>hadoop fs -mkdir /Y21ACB423
```

Output: go to utilities and then click on browse file system

3) List the contents in human readable format:

Syntax: hadoop fs –ls /

Output:

```
E:\hadoop-3.3.6>hadoop fs -ls /
Found 12 items
-rw-r--r--   1 Administrator supergroup          0 2024-08-31 11:55 /1.txt
drwxr-xr-x   - Administrator supergroup          0 2024-08-27 11:06 /CBDS
drwxr-xr-x   - Administrator supergroup          0 2024-08-27 11:05 /bala
drwxr-xr-x   - Administrator supergroup          0 2024-08-31 11:26 /cbds
-rw-r--r--   1 Administrator supergroup        500 2024-09-14 11:31 /input
drwxr-xr-x   - Administrator supergroup          0 2024-09-17 11:26 /input1
drwxr-xr-x   - Administrator supergroup          0 2024-09-17 11:54 /input2
drwxr-xr-x   - Administrator supergroup          0 2024-09-14 11:42 /output
drwxr-xr-x   - Administrator supergroup          0 2024-09-17 11:39 /output1
-rw-r--r--   1 Administrator supergroup          0 2024-08-31 12:35 /sample.txt
drwxr-xr-x   - Administrator supergroup          0 2024-09-14 11:42 /tmp
drwxr-xr-x   - Administrator supergroup          0 2024-08-31 11:13 /y21acb4044

E:\hadoop-3.3.6>
```

4) Upload a file to hdfs:

Syntax: hadoop fs –put <local sys is src> <dest is hdfs>

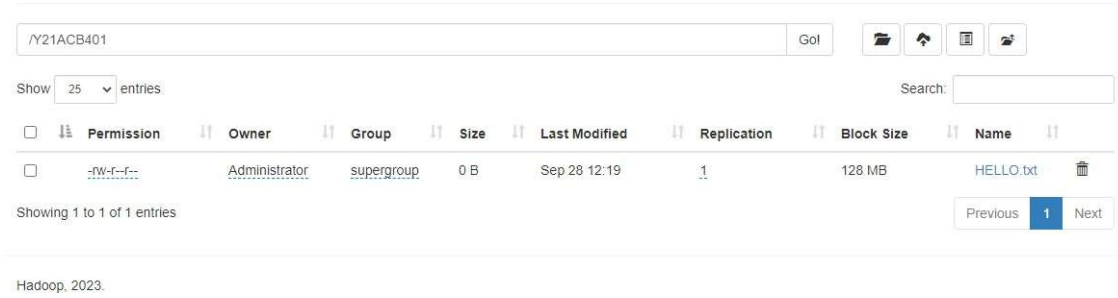
Now we have to create a text file and give some content in it.

```
E:\hadoop-3.3.6>hadoop fs -put C:\HELLO.txt /Y21ACB423

E:\hadoop-3.3.6>\
```

Copy that location (E:\sample.txt) and destination of that file is Y21ACB401 folder in hdfs

Output:



5) View the content of the file:

Syntax: hadoop fs –cat /folder name/filename.txt

Output:

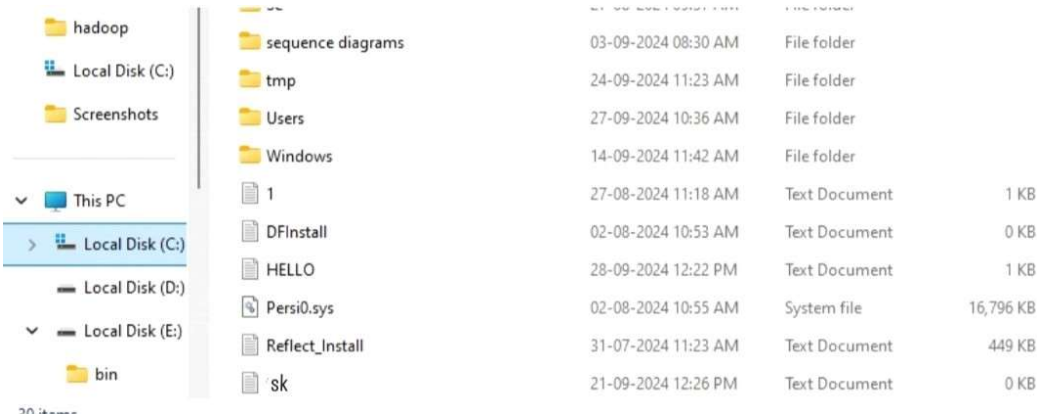
```
E:\hadoop-3.3.6\etc>hadoop fs -cat /Y21ACB423/HELLO.txt
```

6) Download a file from hdfs

Syntax: hadoop fs –get <src is .txt file in hdfs> <dest is local file sys>

```
E:\hadoop-3.3.6>hadoop fs -get /Y21ACB423/HELLO.txt C:\sk.txt
```

Output:

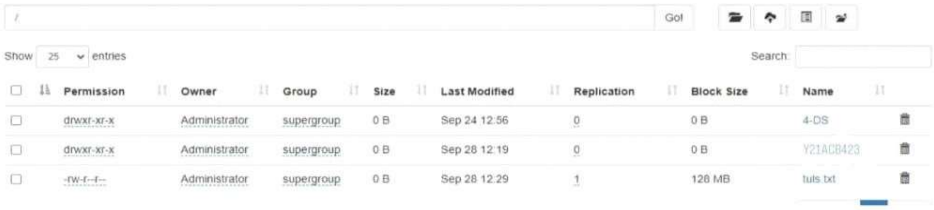


7) Create empty file in hdfs

Syntax: hadoop fs –touchz <filename.txt

```
E:\hadoop-3.3.6>hadoop fs -touchz /tuls.txt
```

Output:



8) Delete a file in hdfs

Syntax:hadoop fs –rm <path>

Output:

```
E:\hadoop-3.3.6>hadoop fs -rm /Y21ACB423/Hello.txt
Deleted /Y21ACB423/Hello.txt
```

9) Moving a file from one folder to another in hdfs

Syntax: hadoop fs –mv <text file folder path > <dest folder>

```
E:\hadoop-3.3.6\etc>hadoop fs -mv /4CBDS/11.txt /Y21ACB423
```

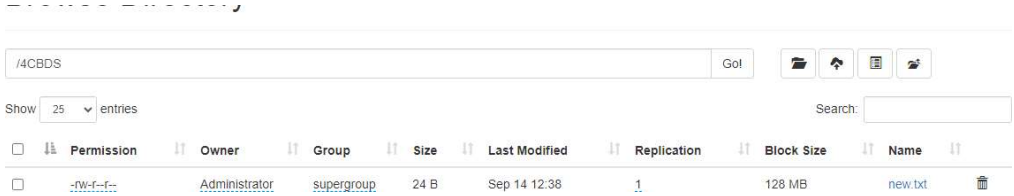
Output:

10) **Copy from local to hdfs:**

Syntax: hadoop fs -copyFromLocal <local file path is the src> <dest is hdfs>

```
E:\hadoop-3.3.6\etc\hadoop>hadoop fs -copyFromLocal E:\new.txt /4CBDS
```

Output:

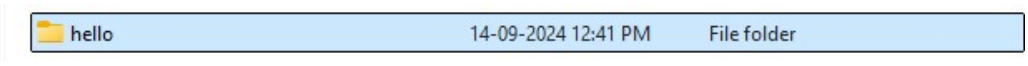


11) copy to local

Syntax:hadoop fs –copyToLocal <src is hdfs> <dest is local >

```
E:\hadoop-3.3.6\etc\hadoop>hadoop fs -copyToLocal /4CBDS E:\hello
```

Output:



12) Count the number of files and bytes under specified path

Syntax: hadoop fs -count

```
E:\hadoop-3.3.6\bin>hadoop fs -count /
           16           15            1710623 /

E:\hadoop-3.3.6\bin>
```

13) To view the last kilo bytes in a file(TAIL):

Syntax: hadoop fs –tail <filename>

Output:

```
E:\hadoop-3.3.6\bin>hadoop fs -tail /Y21ACB423 /BDA.txt
Hello,Welcome to HADOOP lab
E:\hadoop-3.3.6\bin>
```

14) Displaying the total size of a file or directory in HDFS in a human readable format

Syntax: hadoop fs -du -s -h <path>**Output:**

```
E:\hadoop-3.3.6\bin>hadoop fs -du -s -h /Y21ACB423 /BDA.txt
27  27  /Y21ACB423 /BDA.txt

E:\hadoop-3.3.6\bin>
```

15) Changing permissions in HDFS

Syntax: hadoop fs -chmod <permissions> <hdfs-path>

Output:

```
E:\hadoop-3.3.6\bin>hadoop fs -chmod 644 , /Y21ACB423 /sample.txt

E:\hadoop-3.3.6\bin>
```

EXPERIMENT -03

AIM: Write a Map Reduce Program For Word Count.

- Create an Exp3 folder in the Hadoop File System using the command hadoop fs -mkdir /Exp3.

Write it and press and Enter:

```
E:\hadoop-3.3.6\etc\hadoop>hadoop fs -mkdir /Exp3
```

- You can check it using the File System Browser. Open your preferred web browser and enter the address: localhost:9870. Now, click on Utilities > Browse the file system. And then you will see the Exp3 folder listed:

- Now create a text file in your local file system which consists of some data

Now, you need to put this file inside /Exp3 folder created in the Hadoop File System. Go back to the command prompt and type: hadoop fs -put <path_to_txt_file> /Exp3.
where <path_to_txt_file> is the path where your text file is stored. In my case, I have it on E:\exp3_put.txt

command: hadoop fs -put

```
E:\hadoop-3.3.6\etc\hadoop>hadoop fs -put E:\exp3_put.txt /Exp3
```

- you can check the first lines of the .txt file using the head command:

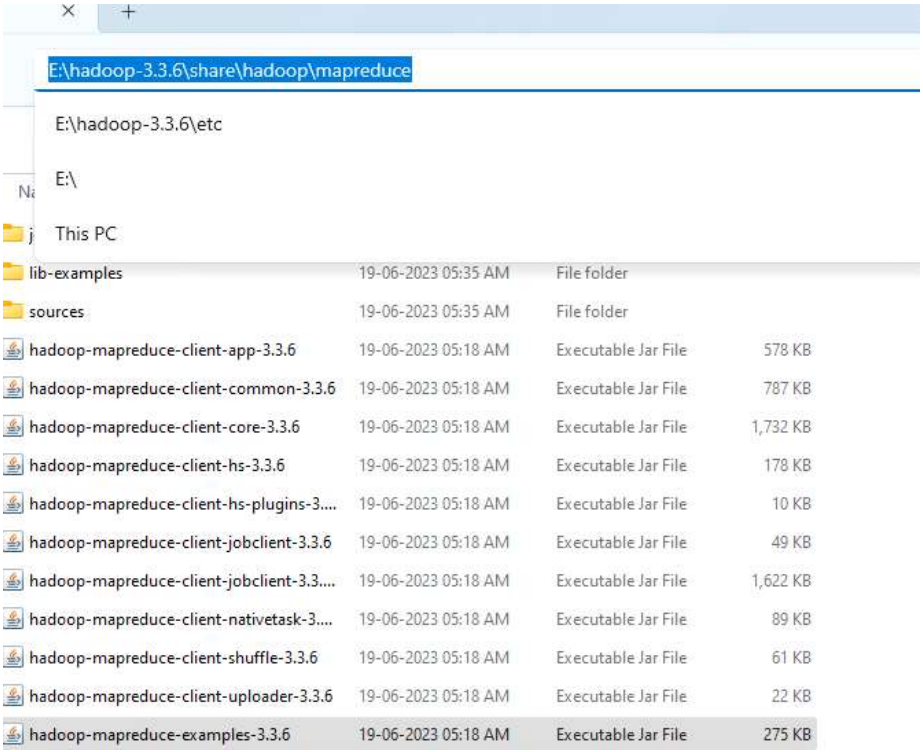
```
E:\hadoop-3.3.6\etc\hadoop>hadoop fs -head /Exp3/exp3_put.txt
Definition. By Mayo Clinic Staff. Headache is pain in any region of the head. Headac
hes may occur on one or both sides of the head, be isolated to a certain location, r
adiate across the head from one point, or have a viselike quality. A headache may ap
pear as a sharp pain, a throbbing sensation or a dull ache.

Common causes
Headaches can have causes that aren't due to underlying disease. Examples include la
ck of sleep, an incorrect eyeglass prescription, stress, loud noise exposure or tigh
t head wear.

Self-treatment
Remedies that may reduce headache pain include aspirin, paracetamol and ibuprofen. R
esting in a darkened room may also help.

Seeking medical care
See a doctor immediately if you:
Feel worse than usual
Get a sudden, severe headache
Become confused, slur your speech or faint
Have one-sided numbness or paralysis, or trouble seeing, speaking or walking
Develop a fever higher than 102°F (39°C)
Experience nausea or vomiting
Make an appointment to see a doctor if you:
Start having f
E:\hadoop-3.3.6\etc\hadoop>
```

- MapReduce is already contained in Hadoop. In my case, it is stored in: E:\hadoop-3.3.6\share\hadoop\mapreduce. You will need the path to the hadoop-mapreduce-examples-3.3.6.jar .jar file in order to run the program. I will use the path E:\hadoop-3.3.6\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar

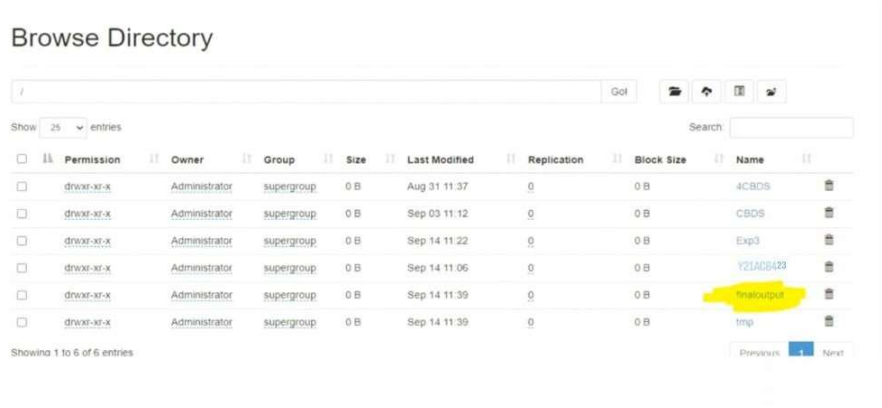


- Now you have located the MapReduce program path, you just need to execute the following command to run it: hadoop jar E:\hadoop-3.3.6\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar wordcount /Exp3 /finaloutput.

which means: "*Run the wordcount program using the content of the /Exp3 folder as the Exp3 store the results in the /finaloutput directory*". You can specify another name for the output folder if you wish. This is a portion of the Command Prompt output and you should see something similar:

```
E:\hadoop-3.3.6\etc\hadoop>hadoop jar E:\hadoop-3.3.6\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar wordcount /Exp3 /finaloutput
2024-09-14 11:39:25,639 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-14 11:39:26,885 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Administrator/.staging/j
726291511089_0001
2024-09-14 11:39:27,692 INFO input.FileInputFormat: Total input files to process : 1
2024-09-14 11:39:28,179 INFO mapreduce.JobSubmitter: number of splits:1
2024-09-14 11:39:28,537 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1726291511089_0001
2024-09-14 11:39:28,538 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-09-14 11:39:28,697 INFO conf.Configuration: resource-types.xml not found
2024-09-14 11:39:28,697 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-09-14 11:39:28,951 INFO impl.YarnClientImpl: Submitted application application_1726291511089_0001
2024-09-14 11:39:29,000 INFO mapreduce.Job: The url to track the job: http://RP23SYS14:8088/proxy/application_1726291511089_0001/
2024-09-14 11:39:29,001 INFO mapreduce.Job: Running job: job_1726291511089_0001
2024-09-14 11:39:37,172 INFO mapreduce.Job: Job job_1726291511089_0001 running in uber mode : false
2024-09-14 11:39:37,174 INFO mapreduce.Job:  map 0% reduce 0%
2024-09-14 11:39:42,291 INFO mapreduce.Job:  map 100% reduce 0%
2024-09-14 11:39:47,345 INFO mapreduce.Job:  map 100% reduce 100%
2024-09-14 11:39:48,357 INFO mapreduce.Job: Job job_1726291511089_0001 completed successfully
2024-09-14 11:39:48,452 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=1789
                FILE: Number of bytes written=558739
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1251
                HDFS: Number of bytes written=1229
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
```

- If you open http://localhost:9870/explorer.html# You will see the final output folder



- Now, if you click on /finaloutput directory you will see its content:

If I open the part-r-00000 file, then I will be able to select if I want to see its head (first) or tail (last) 32 Kb of information. For example, I will take a glimpse to the head of the file:

Browse Directory

/finaloutput Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	Administrator	supergroup	0 B	Sep 14 11:39	1	128 MB	_SUCCESS
-rw-r--r--	Administrator	supergroup	1.2 KB	Sep 14 11:39	1	128 MB	part-r-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information – Block 0

Block ID: 1073741834

Block Pool ID: BP-563988234-172.16.3.114-1725084404877

Generation Stamp: 1010

Size: 1229

Availability:

- RP23SYS14.win2k18.local

File contents

```
the 3
throbbing 1
tight 1
time 1
to 3
trouble 1
underlying 1
usual 1
```

File contents

```
the 3
throbbing 1
tight 1
time 1
to 3
trouble 1
underlying 1
usual 1
```

Close

EXPERIMENT -04

AIM: Steps to Create a Map Reduce Program for Card Count Dataset:

This program will count how many cards of each suit (Hearts, Spades, Diamonds, Clubs) are present in the input data set.

- Create a Java Project
- Write the Mapper and Reducer classes
- Write the Driver (main) class
- Compile the code and package it as a JAR
- Run the JAR on Hadoop.

Java Map Reduce Program for Card Count:

1. Mapper Class:

This class will map each card suit to a count of :

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CardMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text cardSuit = new Text();

    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
        // Split the input line by space
        String[] card = value.toString().split(" ");

        if (card.length == 2) {
            // The first part is the card suit
            cardSuit.set(card[0]);

            // Emit the suit and a count of 1
            context.write(cardSuit, one);
        }
    }
}
```

2. Reducer Class:

The reducer will sum up the counts for each card suit.

java

Copy code

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class CardReducer extends Reducer<Text, IntWritable, Text,
IntWritable> {

  public void reduce(Text key, Iterable<IntWritable> values, Context
context) throws IOException, InterruptedException {
        int sum = 0;

        // Sum up the values (card counts)
        for (IntWritable val : values) {
            sum += val.get();
        }

        // Emit the suit and the sum
        context.write(key, new IntWritable(sum));
    }
}
```

3. Driver Class:

This class is the main entry point for the MapReduce job.

```
java
Copy code
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CardCount
{

    public static void main(String[] args) throws Exception
        {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Card Count");

        // Set the jar class
        job.setJarByClass(CardCount.class);

        // Set Mapper and Reducer classes
        job.setMapperClass(CardMapper.class);
        job.setReducerClass(CardReducer.class);

        // Set the output key and value types
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        // Input and output paths from command line
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        // Exit the program after job completion
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

4. Compiling and Creating the JAR

- First, ensure you have Hadoop libraries added to your project's classpath.
- Compile the Java classes:

```
bash
Copied!
javac -classpath `hadoop classpath` -d . CardMapper.java
CardReducer.java CardCount.java
```

Package the compiled files into a JAR file:

```
jar -cvf cardcount.jar -C .
```

5. Running the Program on Hadoop

- **Prepare Input Data**: Place your input data (e.g., `card_data.txt`) in HDFS:

```
hdfs dfs -mkdir /user/hadoop/card_input
hdfs dfs -put /local/path/to/card_data.txt /user/hadoop/card_input
```

Run the MapReduce Job:

```
hadoop jar cardcount.jar CardCount /user/hadoop/card_input
/user/hadoop/card_output
```

Check Output: Once the job completes, check the output in HDFS:

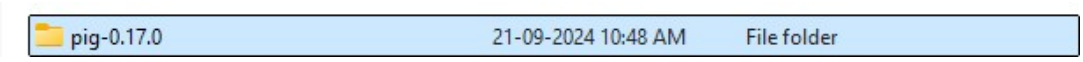
```
hdfs dfs -cat /user/hadoop/card_output/part-r-00000
```

EXPERIMENT -05

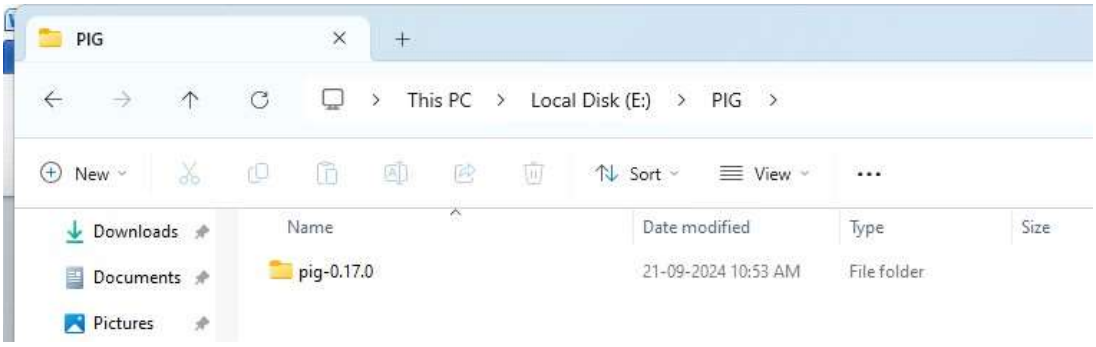
AIM: PIG Installation

Extract the above file using winrar (extract to pig-0.17.0\)

And once again extract it click on yes to all.Then you will the below folder in downloads

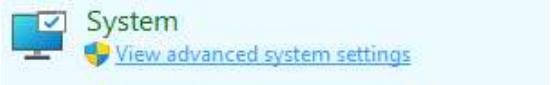


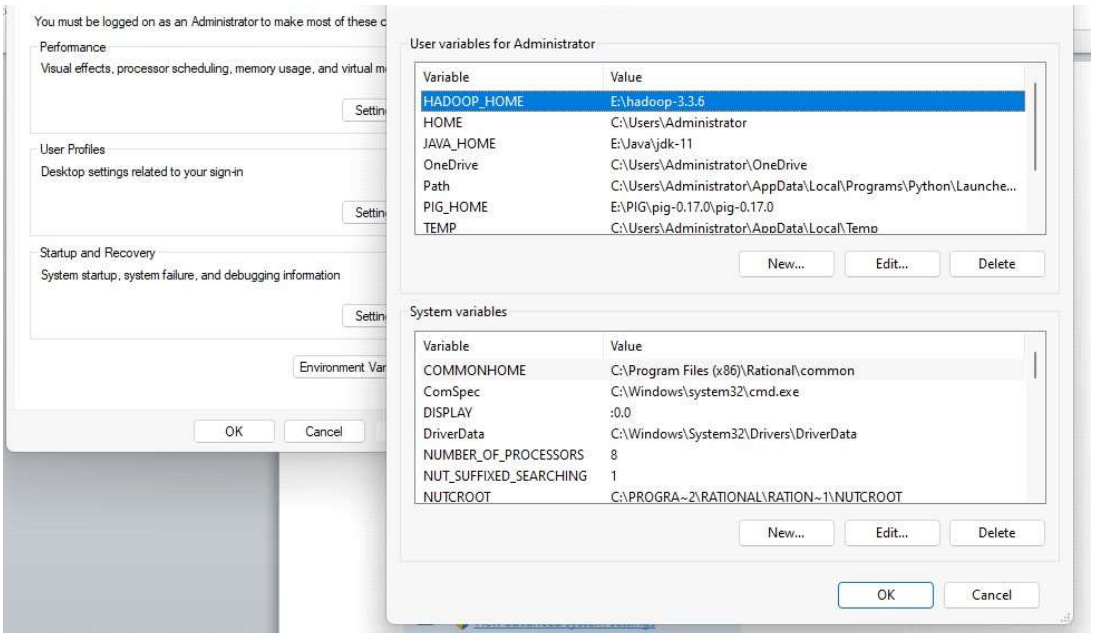
Now we can organize our PIG installation, we can create a folder and move the final extracted file in it.(E:\PIG)



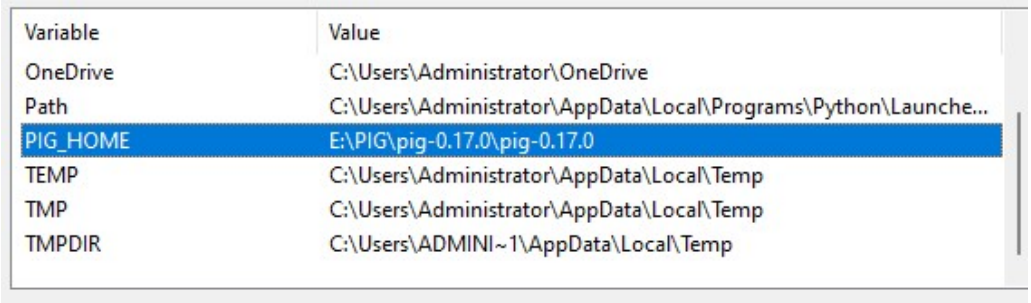
- Please note while creating folders, DO NOT ADD SPACES IN BETWEEN THE FOLDER NAME.(it can cause issues later)

- Go to Control Panel > System > click on the "Advanced system settings" link to edit environment variables.



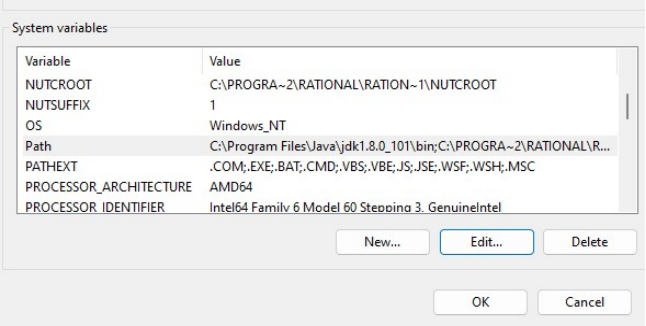


- Open environment Variable and click on "New" in "User Variable".



Variable	Value
OneDrive	C:\Users\Administrator\OneDrive
Path	C:\Users\Administrator\AppData\Local\Programs\Python\Launche...
PIG_HOME	E:\PIG\pig-0.17.0\pig-0.17.0
TEMP	C:\Users\Administrator\AppData\Local\Temp
TMP	C:\Users\Administrator\AppData\Local\Temp
TMPDIR	C:\Users\ADMINI~1\AppData\Local\Temp

- Select Path variable in the system variables and click on “Edit



System variables

Variable	Value
NUTCROOT	C:\PROGRA~2\RATIONAL\RATION~1\NUTCROOT
NUTSUFFIX	1
OS	Windows_NT
Path	C:\Program Files\Java\jdk1.8.0_101\bin;C:\PROGRA~2\RATIONAL\R...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 60 Stepping 3, GenuineIntel

New... Edit... Delete

OK Cancel

E:\hadoop-3.3.6\bin
E:\hadoop-3.3.6\sbin
E:\PIG\pig-0.17.0\pig-0.17.0\bin

OK

- Click OK and OK. & we are done with Setting Environment Variables.

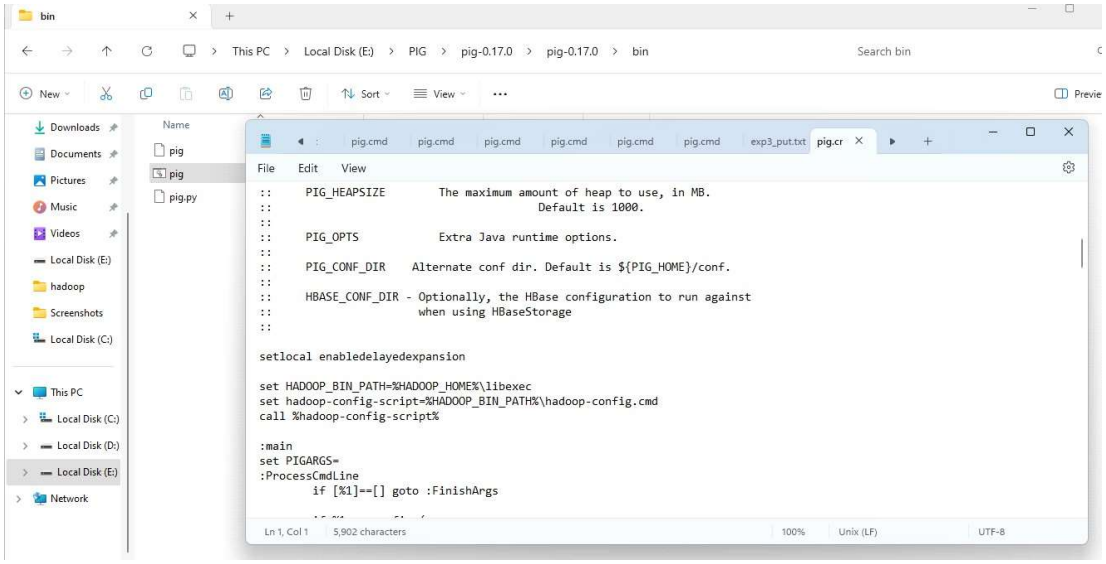
- Open the pig. cmd file in edit mode.

  We can find the file in the bin folder.

- Now we need to change the value of the HADOOP_BIN_PATH

- Old value:- %HADOOP_HOME%\bin

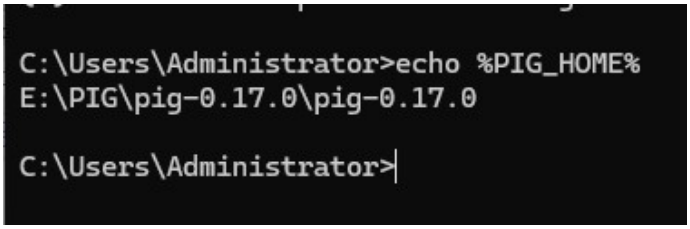
  `New Value:- %HADOOP_HOME%\libexec`



Verify the Paths

- Now we need to verify that what we have done is correct and reflecting.
- Open a **NEW** Command Window
- Run following commands

```
echo %PIG_HOME%
```



Save the file.

The next step is to verify the setup once again. So, we need to execute the

`pig -version` command once again.

```
C:\Users\Administrator>pig -version
Apache Pig version 0.17.0 (r1797386)
compiled Jun 02 2017, 15:41:58
```

Starting PIG

Now we need to start a new Command Prompt remember to run it as administrator to avoid permission issues and execute the below commands pig

```
C:\Users\Administrator>pig
2024-09-28 12:18:09,033 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2024-09-28 12:18:09,045 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2024-09-28 12:18:09,045 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2024-09-28 12:18:09,172 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15
:41:58
2024-09-28 12:18:09,172 [main] INFO  org.apache.pig.Main - Logging error messages to: E:\hadoop-3.3.6\logs\pig_172750608
9156.log
2024-09-28 12:18:09,198 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file C:\Users\Administrator/.pigboo
tup not found
2024-09-28 12:18:09,478 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated
. Instead, use mapreduce.jobtracker.address
2024-09-28 12:18:09,478 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hado
op file system at: hdfs://localhost:9000
2024-09-28 12:18:09,914 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-8927652c-6674
-406a-9323-87e16e53c944
2024-09-28 12:18:09,914 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set
to false
grunt>
```

EXPERIMENT- 06

AIM: Write the word count script using pig latin.

Steps for pig word count:

- Load the input file: The first step is to load the input text file from HDFS (or) local storage.
- Split the lines into words: You need to split each line of text into individual words.
- Group by word: Group all occurrences of each word.
- Count the occurrences: Use pig's built in function to count the number of times each word appears.
- Store the result: Save (or) display the result.

one	01-10-2024 11:17 AM	Text Document	1 KB
sample1	17-09-2024 11:23 AM	Text Document	1 KB
wordcount_pig	05-10-2024 10:43 AM	Text Document	0 KB

(C:)
(D:)

Explanation of each step:

```
starting yarn daemons
E:\hadoop-3.3.6\etc>hadoop fs  -put E:\wordcount_pig.txt /y21acb423
```

Wordcount_pig.txt file loaded from local system to HDFS.

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	Administrator	supergroup	85 B	Oct 05 10:55	1	128 MB	wordcount_pig.txt

Showing 1 to 1 of 1 entries Previous 1 Next

Run pig then we get "grunt" shell.

```
2024-10-05 10:56:04,265 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file C:\Users\Administrator/.pigbootup not found
2024-10-05 10:56:04,661 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.job
tracker.address
2024-10-05 10:56:04,661 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://lo
calhost:9000
2024-10-05 10:56:05,105 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-c65881aa-e0a5-4276-8bbd-16c64395c2cf
2024-10-05 10:56:05,105 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt>
```

- **LOAD:** Loads the input file into the lines relation, where each record is a line from text file. To load the input file use the following command:

  input_lines =LOAD 'text file path' AS (line: chararray);

```
Details at logfile: D:\hadoop-3.3.6\logs\pig_1728107221751.log
grunt> input_lines = LOAD ' /y21acb401/wordcount_pig.txt' AS (line: chararray);
grunt> dump input_lines
2024-10-05 11:22:14,578 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig featur
```

Checkout the file loaded or not by using the following command:

grunt> **dump input_lines**

Output:

```
AP_ONLY hdfs://localhost:9000/tmp/temp-367973451/tmp1861263690,

Input(s):
Successfully read 0 records from: "/y21acb423/wordcount_pig.txt"

Output(s):
Successfully stored 0 records in: "hdfs://localhost:9000/tmp/temp-367973451/tmp1861263690"

Counters:
Total records written : 0
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1728107190646_0001
```

```
not generate code.
2024-10-05 11:30:41,394 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input fil
s : 1
2024-10-05 11:30:41,394 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total i
o process : 1
(this pig word count )
(pig is a tool in bda)
(pig does the segregation of word count )
grunt>
```

- **TOKENIZE:** Splits each line into individual words. TOKENIZE function returns a bag of words, and FLATTEN converts each word into separate rows. Use the following command to tokenize:

  grunt> **words = FOREACH record GENERATE FLATTEN (TOKENIZE(line)) AS word;**

```
grunt>
grunt> words =FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;
grunt>
```

  grunt> **dump words;**

```
grunt> dump words;
```

```
s : 1
2024-10-05 11:46:55,677 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths t
o process : 1
(this)
(pig)
(word)
(count)
(pig)
(is)
(a)
(tool)
(in)
(bda)
(pig)
(does)
(the)
(segregation)
(of)
(word)
(count)
grunt>
```

- **GROUP:** Groups all records by each unique word. Use the following command to group:

grunt> **grouped = GROUP words BY word;**

grunt**> dump grouped**

```
(segregation,{(segregation)})
grunt> wordcount = FOREACH grouped GENERATE group, COUNT(words);
grunt> dump
wordcount     input_lines   grouped       words
grunt> dump wordcount;
2024-10-05 12:01:11,688 [main] INFO  org.apache.pig.tools.pigstats.ScriptS
```

Output:

```
not generate code.
2024-10-05 11:58:04,014 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to proces
s : 1
2024-10-05 11:58:04,025 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths t
o process : 1
(a,{(a)})
(in,{(in)})
(is,{(is)})
(of,{(of)})
(bda,{(bda)})
(pig,{(pig),(pig),(pig)})
(the,{(the)})
(does,{(does)})
(this,{(this)})
(tool,{(tool)})
(word,{(word),(word)})
(count,{(count),(count)})
(segregation,{(segregation)})
grunt>
Words: 0   English (India)
```

- **FILTER:** This step is optional but filters out null (or) empty words, which may occur due to multiple spaces or other reasons.
- **COUNT:** For each group (i.e.,each word)), count the number of occurrences.

grunt> **wordcount = FOREACH grouped GENERATE group, COUNT(words);**

```
grunt> wordcount = FOREACH  grouped GENERATE group, COUNT(words);
grunt>
```

grunt> dump wordcount;

```
grunt> dump wordcount;
```

```
2024-10-05 12:09:42,212 [main] INFO  org.apache.pig.data.S
2024-10-05 12:09:42,238 [main] INFO  org.apache.hadoop.map
2024-10-05 12:09:42,238 [main] INFO  org.apache.pig.backer
(a,1)
(in,1)
(is,1)
(of,1)
(bda,1)
(pig,3)
(the,1)
(does,1)
(this,1)
(tool,1)
(word,2)
(count,2)
(segregation,1)
grunt> |
```

6. **STORE:** Store the results into the word_count.

```
(segregation,1)
grunt> STORE wordcount INTO 'wordcount_output';
2024-10-05 12:12:51,388 [main] INFO  org.apache.hadoop.conf.Configuration
recated. Instead, use yarn.system-metrics-publisher.enabled
2024-10-05 12:12:51,404 [main] INFO  org.apache.hadoop.conf.Configuration
e mapreduce.output.textoutputformat.separator
2024-10-05 12:12:51,435 [main] INFO  org.apache.pig.tools.pigstats.Script
```

QUIT GRUNT

7. **pig -x local wordcount.pig**

```
E:\hadoop-3.3.6\etc>pig -x local wordcount.pig
2024-10-05 12:25:22,379 INFO pig.ExecTypeProvider: Trying Exe
2024-10-05 12:25:22,379 INFO pig.ExecTypeProvider: Picked LOC
```

8. **Dispalys the content of the file**

cat wordcount_output/part-r-00000

```
E:\hadoop-3.3.6\etc>hadoop fs -cat wordcount_output/part-r-00000
a       1
in      1
is      1
of      1
bda     1
pig     3
the     1
does    1
this    1
tool    1
word    2
count   2
segregation     1
```