Assignment Solution

Week2: MapReduce - Distributed Computing Framework

TRENDYTECH 9108179578

Week 2-Assignment Solutions

Total Marks 30

Note: Question 1 to 5 are optional

We have an input directory named **input_data**, which contains two text files:

Qu 1) Copy the input_data directory to a directory named mapred_input residing in home directory of hdfs.

Display contents of **mapred_input** directory in a single command and also the contents of the files copied. Share the snapshot of the same.

TRENDYTECH
UPLIFT YOUR CAREER!

Solution:

1. Create a directory in hdfs home directory:

```
hadoop fs -mkdir /user/cloudera/mapred_input
```

Copying the files from local to hdfs

Using Absolute path:

```
hadoop fs -put /home/cloudera/Desktop/input_data
/user/cloudera/mapred_input/
```

TRENDYTECH 9108179578

or

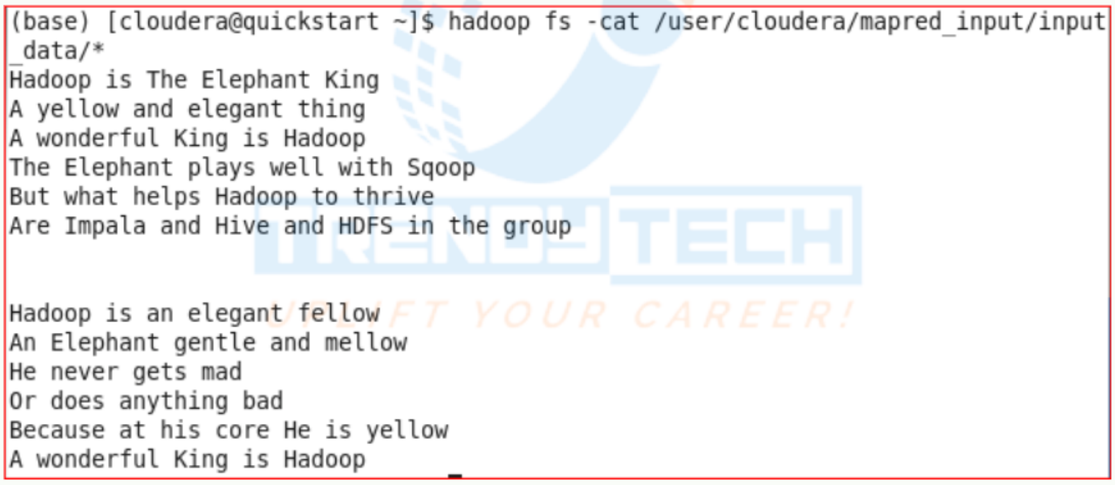
Using relative path:

hadoop fs -put Desktop/input_data   mapred_input/

Display contents inside **mapred_input** directory in a single command and also the contents of the files copied. Share the snapshot of the same.

hadoop fs -ls -R /user/cloudera/mapred_input

hadoop fs -cat /user/cloudera/mapred_input/input_data/*



```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/mapred_input/input
_data/*
Hadoop is The Elephant King
A yellow and elegant thing
A wonderful King is Hadoop
The Elephant plays well with Sqoop
But what helps Hadoop to thrive
Are Impala and Hive and HDFS in the group


Hadoop is an elegant fellow
An Elephant gentle and mellow
He never gets mad
Or does anything bad
Because at his core He is yellow
A wonderful King is Hadoop
```

Qu 2)

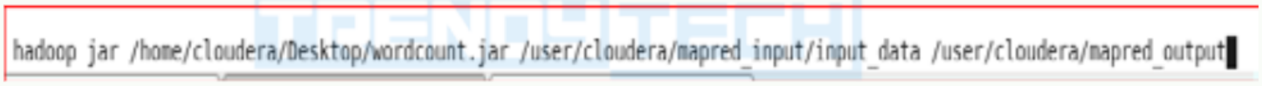
- Write and execute a wordcount program to count the frequency for each word in these two input files.

TRENDYTECH 9108179578

- Create a runnable jar file named **wordcount.jar** from the above code and execute that jar on the given input files, the output should be generated in a directory called **mapred_output** inside home directory of hdfs.
- Share the snapshot of the command used to run the jar file in terminal.
- Display the contents of the **mapred_output** directory and the reducer part file generated. Share the snapshot of the same.

Solution:

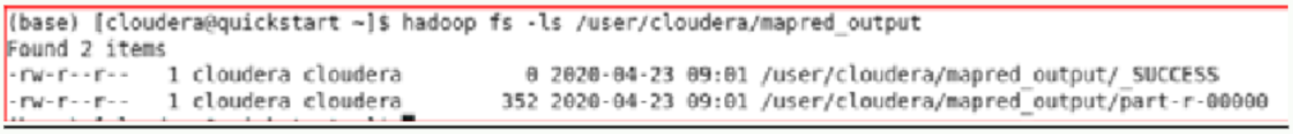
Share the snapshot of the command to run the jar file in terminal.



```
hadoop jar /home/cloudera/Desktop/wordcount.jar /user/cloudera/mapred_input/input_data /user/cloudera/mapred_output
```

hadoop jar /home/cloudera/Desktop/wordcount.jar /user/cloudera/mapred_input/input_data /user/cloudera/mapred_output

Display the contents of the **mapred_output** directory. Share the snapshot of the same



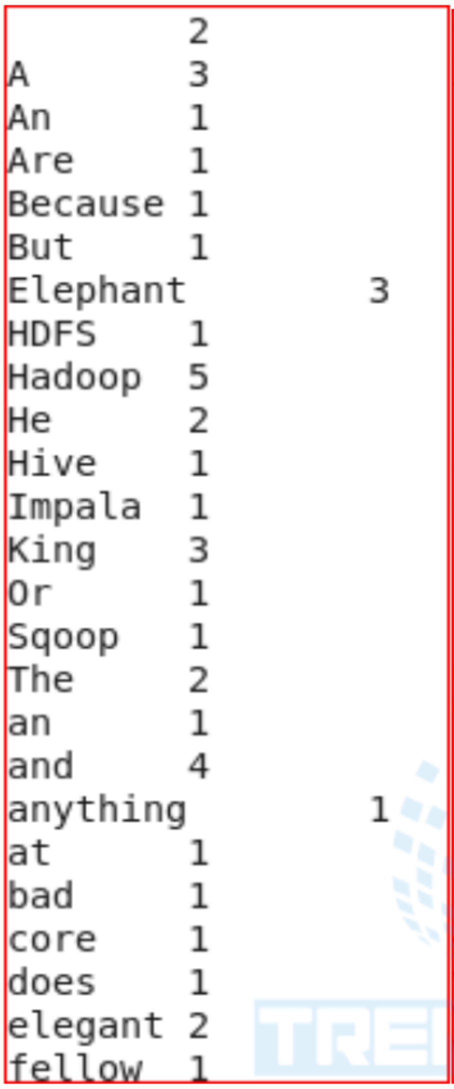
```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/mapred_output
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2020-04-23 09:01 /user/cloudera/mapred_output/_SUCCESS
-rw-r--r--   1 cloudera cloudera        352 2020-04-23 09:01 /user/cloudera/mapred_output/part-r-00000
```

hadoop fs -cat /user/cloudera/mapred_output/part-r-00000

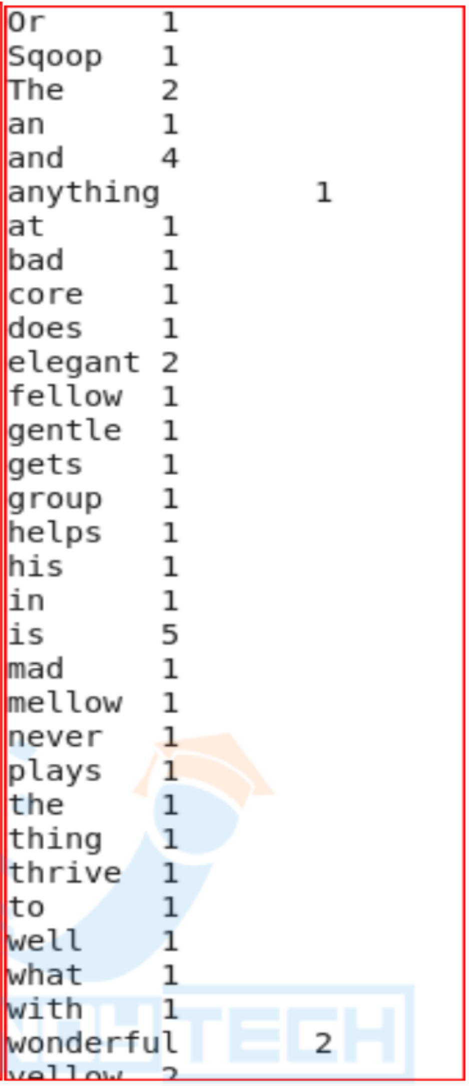
Contents of the output part file generated:

TRENDYTECH UPLIFT YOUR CAREER!

TRENDYTECH 9108179578



```
        2
A       3
An      1
Are     1
Because 1
But     1
Elephant        3
HDFS    1
Hadoop  5
He      2
Hive    1
Impala  1
King    3
Or      1
Sqoop   1
The     2
an      1
and     4
anything        1
at      1
bad     1
core    1
does    1
elegant 2
fellow  1
```



```
Or      1
Sqoop   1
The     2
an      1
and     4
anything        1
at      1
bad     1
core    1
does    1
elegant 2
fellow  1
gentle  1
gets    1
group   1
helps   1
his     1
in      1
is      5
mad     1
mellow  1
never   1
plays   1
the     1
thing   1
thrive  1
to      1
well    1
what    1
with    1
wonderful       2
yellow  2
```

TRENDYTECH
UPLIFT YOUR CAREER!

Qu 3) What change will you make in the above code if we do not want any aggregation finally.**Just mention the change in the code.**

Solution:

In the Main.java file just set number of Reducers to 0

job.setNumReduceTasks(0);

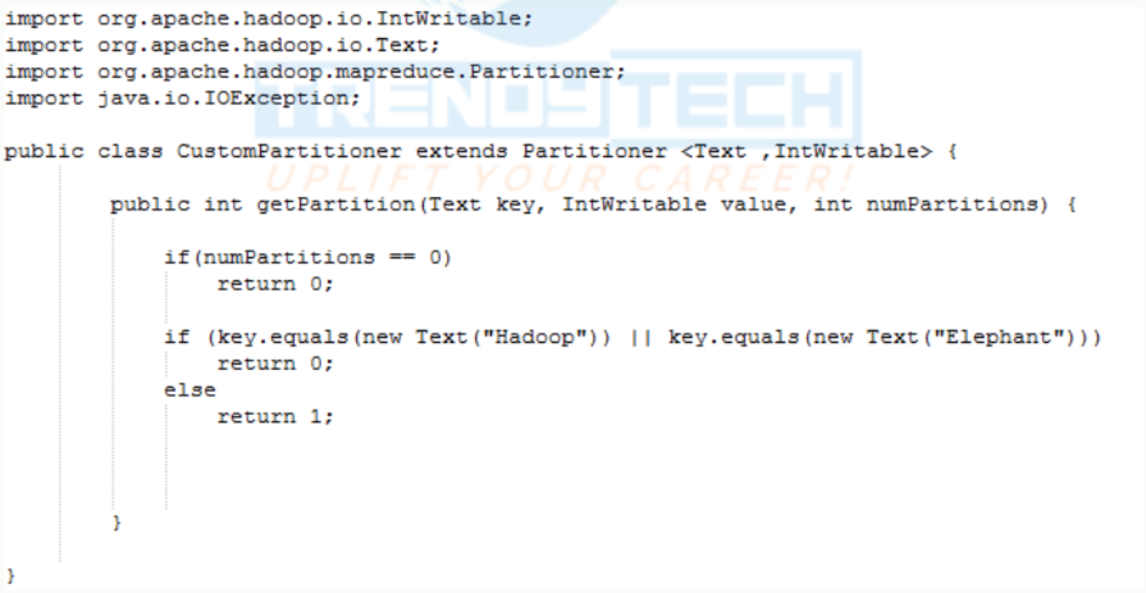
TRENDYTECH 9108179578

Qu 4) Write the code if we want the words present in the input files - **Hadoop , Elephant** to go to one reducer and the other remaining words to go to the second reducer.

- Create a runnable jar file **wc_part.jar ,**execute the jar file , Share the snapshot of the command
- Place the output in **wc_part_out** directoryinside home directory of hdfs. Share the snapshot of the output generated.

Solution:

CustomPartitioner.java

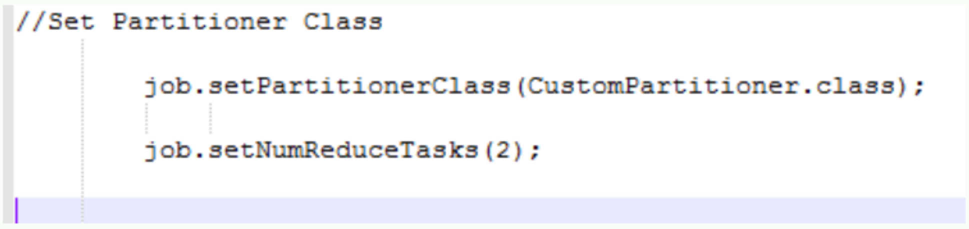


```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Partitioner;
import java.io.IOException;

public class CustomPartitioner extends Partitioner <Text ,IntWritable> {

        public int getPartition(Text key, IntWritable value, int numPartitions) {

            if(numPartitions == 0)
                return 0;

            if (key.equals(new Text("Hadoop")) || key.equals(new Text("Elephant")))
                return 0;
            else
                return 1;


        }

}
```

Main.java

Make these changes in the Main.java code. CustomPartitioner is a user defined class.

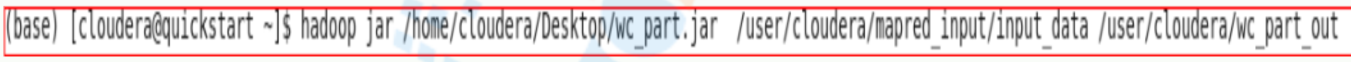
TRENDYTECH
UPLIFT YOUR CAREER!
TRENDYTECH 9108179578



```
//Set Partitioner Class

        job.setPartitionerClass(CustomPartitioner.class);

        job.setNumReduceTasks(2);
```

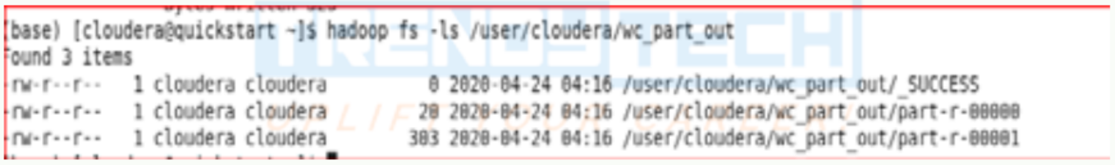
Create a runnable jar file **wc_part.jar** ,execute the jar file , and then place the output in **wc_part_out** directoryinside home directory of hdfs. Share the snapshot of the command and also snapshot of the output.

hadoop jar /home/cloudera/Desktop/wc_part.jar /user/cloudera/mapred_input/input_data /user/cloudera/wc_part_out



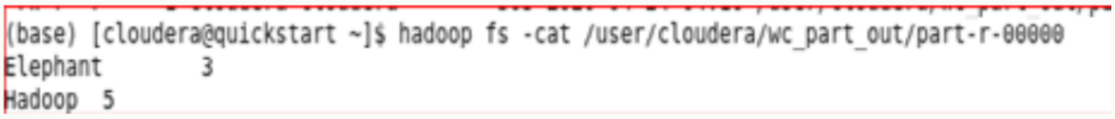
```
(base) [cloudera@quickstart ~]$ hadoop jar /home/cloudera/Desktop/wc_part.jar  /user/cloudera/mapred_input/input_data /user/cloudera/wc_part_out
```

hadoop fs -ls /user/cloudera/wc_part_out



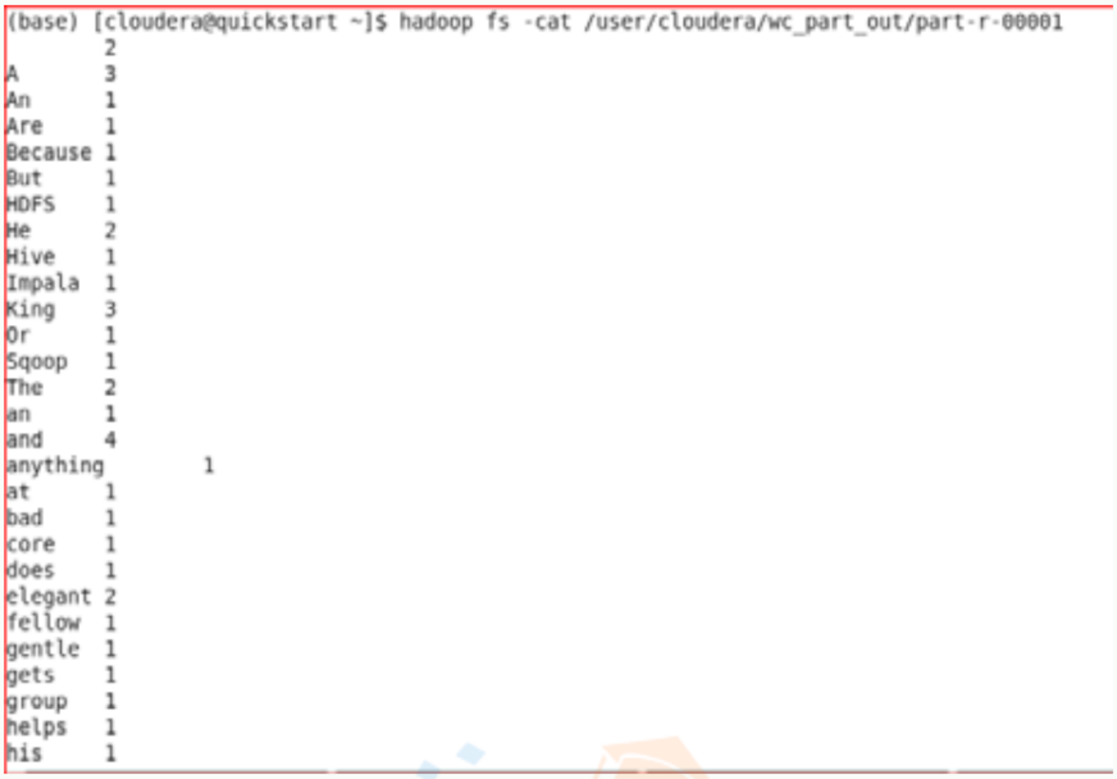
```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/wc_part_out
Found 3 items
-rw-r--r--   1 cloudera cloudera          0 2020-04-24 04:16 /user/cloudera/wc_part_out/_SUCCESS
-rw-r--r--   1 cloudera cloudera         20 2020-04-24 04:16 /user/cloudera/wc_part_out/part-r-00000
-rw-r--r--   1 cloudera cloudera        303 2020-04-24 04:16 /user/cloudera/wc_part_out/part-r-00001
```

Output Snapshot:

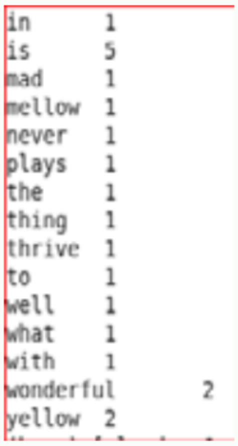


```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wc_part_out/part-r-00000
Elephant        3
Hadoop  5
```

TRENDYTECH 9108179578



```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wc_part_out/part-r-00001
        2
A       3
An      1
Are     1
Because 1
But     1
HDFS    1
He      2
Hive    1
Impala  1
King    3
Or      1
Sqoop   1
The     2
an      1
and     4
anything        1
at      1
bad     1
core    1
does    1
elegant 2
fellow  1
gentle  1
gets    1
group   1
helps   1
his     1
```



```
in      1
is      5
mad     1
mellow  1
never   1
plays   1
the     1
thing   1
thrive  1
to      1
well    1
what    1
with    1
wonderful       2
yellow  2
```

TRENDY TECH
UPLIFT YOUR CAREER!

Qu 5) Write the code : If

a)the key-length <3 ,output should go to reducer 1

b) the key-length = 3,output should go to reducer 2

c) the key-length >3,output should go to reducer 3

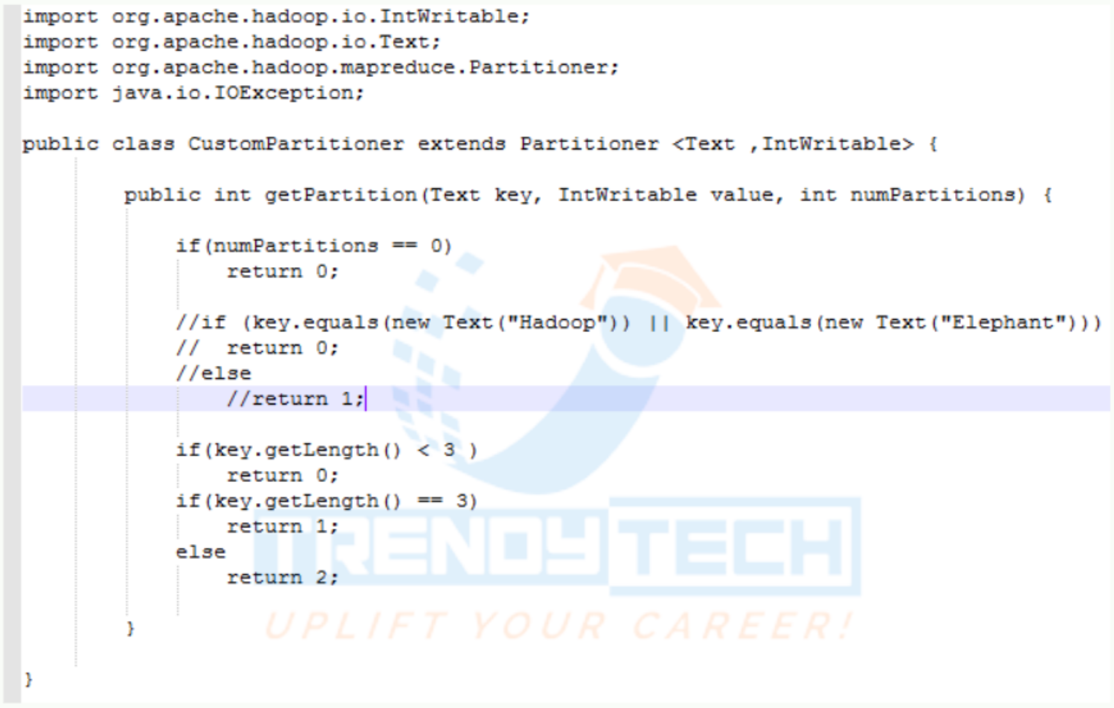
Note: You can comment the previous code of Question 4 and write the logic there itself.

TRENDYTECH 9108179578

- Create a Runnable jar file **wc_custom.jar** ,The output directory should be named **wc_custom_dir,** in hdfs. Execute the jar, share the snapshot of the command for running the jar in terminal and the output snapshot.

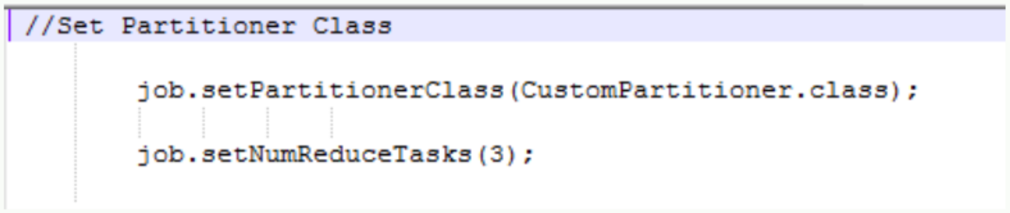
Solution:

Code: CustomPartitioner.java



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

public class CustomPartitioner extends Partitioner <Text ,IntWritable> {

        public int getPartition(Text key, IntWritable value, int numPartitions) {

            if(numPartitions == 0)
                return 0;

            //if (key.equals(new Text("Hadoop")) || key.equals(new Text("Elephant")))
            //  return 0;
            //else
                //return 1;

            if(key.getLength() < 3 )
                return 0;
            if(key.getLength() == 3)
                return 1;
            else
                return 2;

        }

}
```

Changes to be made in Main.java



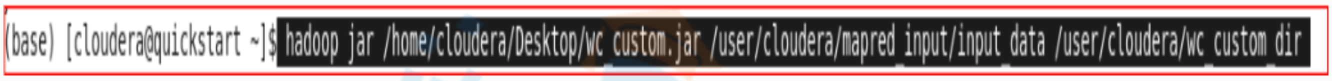
```
//Set Partitioner Class

      job.setPartitionerClass(CustomPartitioner.class);

      job.setNumReduceTasks(3);
```

TRENDYTECH 9108179578

Create a Runnable jar file **wc_custom.jar** ,The output directory should be named **wc_custom_dir,** in hdfs. Execute the jar, share the snapshot of the command and the output snapshot.

hadoop jar /home/cloudera/Desktop/wc_custom.jar /user/cloudera/mapred_input/input_data /user/cloudera/wc_custom_dir

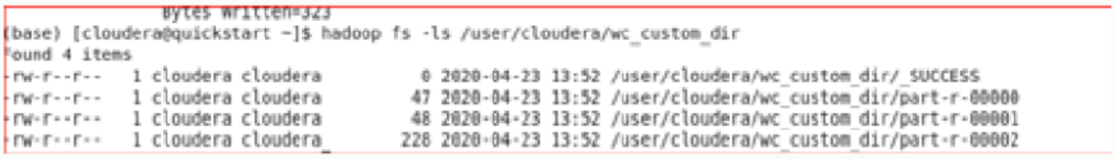
JAR execution in terminal



```
(base) [cloudera@quickstart ~]$ hadoop jar /home/cloudera/Desktop/wc_custom.jar /user/cloudera/mapred_input/input_data /user/cloudera/wc_custom_dir
```

TRENDYTECH
UPLIFT YOUR CAREER!

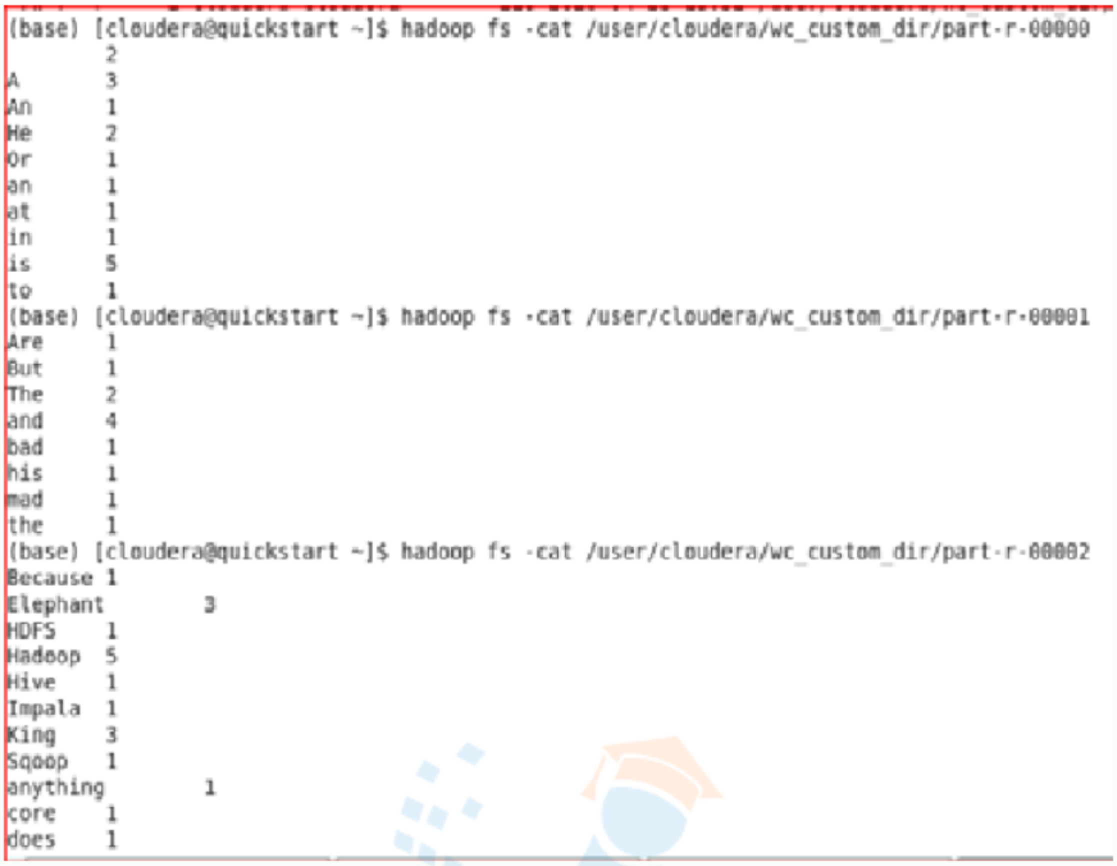
Contents of Output Directory:



```
                Bytes Written=323
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/wc_custom_dir
Found 4 items
-rw-r--r--   1 cloudera cloudera          0 2020-04-23 13:52 /user/cloudera/wc_custom_dir/_SUCCESS
-rw-r--r--   1 cloudera cloudera         47 2020-04-23 13:52 /user/cloudera/wc_custom_dir/part-r-00000
-rw-r--r--   1 cloudera cloudera         48 2020-04-23 13:52 /user/cloudera/wc_custom_dir/part-r-00001
-rw-r--r--   1 cloudera cloudera        228 2020-04-23 13:52 /user/cloudera/wc_custom_dir/part-r-00002
```

Displaying the contents of the output part files:

TRENDYTECH 9108179578


```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wc_custom_dir/part-r-00000
        2
A       3
An      1
He      2
Or      1
an      1
at      1
in      1
is      5
to      1
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wc_custom_dir/part-r-00001
Are     1
But     1
The     2
and     4
bad     1
his     1
mad     1
the     1
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wc_custom_dir/part-r-00002
Because 1
Elephant        3
HDFS    1
Hadoop  5
Hive    1
Impala  1
King    3
Sqoop   1
anything        1
core    1
does    1
```

TRENDYTECH
UPLIFT YOUR CAREER!

Qu 6) (2 Marks)

For the above program, what will happen if you use 3 reducers and in partitioner class you have below condition:

if key length less than 4 than return 0

else return 1

Please explain.

TRENDYTECH 9108179578

Solution:

In the partitioner class we have only **two** return values, that is **0 and 1**. But we have mentioned **3 reducers** in our main function. Therefore, the third reducer won't get any key value pair to work on. **Under utilisation** of reducer is what is going to take place.

Qu 7) what will happen if you use 2 reducers and in partitioner class you have below condition: **(2 Marks)**

if key length less than 4 - than return 0

if key length >= 4 and <6 return 1

else return 2

Please explain.

Solution: This will lead to error.

TRENDYTECH
UPLIFT YOUR CAREER!

Qu 8) For word count problem, in your reducer if the code is

```
long count = 0;
for (IntWritable value : values) {
count = count+1;
}
context.write(key, new LongWritable(count));
```

A) Do you expect correct output if you run this code without combiner & why. please explain. **(2 Marks + 2 Marks)**

B) Do you expect correct output if you run this code with a combiner class & why. please explain **(2 Marks + 2 Marks)**

TRENDYTECH 9108179578

C) In above problem how will you make sure that output is correct along with the right optimization. What changes will you make. **(2 Marks + 2 Marks**)

Solution:

A) **Yes, we would get the correct output** if we have this as a reducer code,even if we are using no combiners. After shuffle and sort the value that we would get as input is (<key>,{1,1,1,1,...,1}). So, basically we are counting the no. of ones, or the no. of elements present in the list.

B) **No, we would not get the correct output in this case**. As, combiner would do some local aggregations, therefore now the key value pair wouldn't be (<key>,1). But would be (<key>,some_no). Where some_no is the no. of occurrences of that particular key in that particular mapper. And now after shuffle and sort, the reducer gets something like (<key>,{3,5,8,2,7,..}) as the input, and the correct output would be sum of all integers in the list and not, the length of the list.

TRENDYTECH UPLIFT YOUR CAREER!

C) I would use a **combiner** which would take care of local aggregations. And then as reducer code would use **count+=value. Instead of count+=1;**

Qu 9) (2 Marks)

A) What can be the use case when reducer is not required. Please explain one such use case.

TRENDYTECH 9108179578

Solution: When we are doing some filtering of data. Like, removing all nos. and punctuations from a file.

B) Is it good in terms of performance if reducer is not required?

(2 Marks)

Solution: Yes,because we can have all work done in Parallel then.Also no shuffle and sort needed.So no data transfer involved from mapper to reducer machine.

C)Will shuffle and sort come into play when there is no reducer? Please explain why? **(2 Marks)**

Solution:

No, there won't be any shuffle & sort when there are no reducers, and mapper's output will be the final one. Shuffle & sort is only required to feed in data to the reducer in a key and a list of values pair format. But since there is no reducer,mapper's output is the final one.

Qu 10) **(4 Marks + 4 Marks)**

In Java:

The **java.lang.Math.random()** is used to return a pseudo random double type number greater than or equal to 0.0 and less than 1.0. The default random number is always generated between 0 and 1.

TRENDYTECH UPLIFT YOUR CAREER!

TRENDYTECH 9108179578

If you want a specific range of values, you have to multiply the returned value with the magnitude of the range. For example, if you want to get the random number between 0 to 20, the resultant has to be multiplied by 20 to get the desired result.

In word count problem ,Consider you are using 2 reducers and we have written the custom partitioning logic as below:

```
if (key.length() + Math.random()*5 < 5)
return 0;
```

What is the behaviour of above code. Please explain what do you feel and why? Do you suggest any changes in the above code ?

TRENDYTECH
UPLIFT YOUR CAREER!

Solution:

The above code is not consistent.

If my random function generate a value 0.1 then random()*5 = 0.5

But of my random function generates a value 0.9 then random()*5 = 4.5

So, suppose my key is of length 3.

Case 1: Random function generated 0.5

3 + 0.5 = 3.5 which is less then 5 so it returns zero

Case 2: Random function generated 0.9

3 + 4.5 = 7.5 which is greater than 5, so it returns 1

TRENDYTECH 9108179578

Hence non consistent.

Changes

A better approach would be to not design a custom partitioner at all and let the system hash function take care.

But if we see 1 reducer is getting more load, i.e., if we see that the load is not balanced, then change the if condition from

if(key.length + Math.random()*5 < 5)

to

if(key.length < 5)

****

TRENDYTECH
UPLIFT YOUR CAREER!

TRENDYTECH 9108179578