Assignment -Week 2

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.

Qu 2)

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

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.**

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** directory inside home directory of hdfs. Share the snapshot of the output generated.

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.

- 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.

Qu 6)

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.

Qu 7) what will happen if you use 2 reducers and in paritioner class you have below condition:

if key length less than 4 - than return 0

if key length >= 4 and <6 return 1

else return 2

Please explain.

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.

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

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

Qu 9)

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

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

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

Qu 10)

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.

If you want to get 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;

else
return 1;
```

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

****