Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Big Data Systems – Spark Lab Sheet: 2

Word Count with Spark

1. Objective

Students should be able to

A. Get familiarity with the execution of Python programs on the Spark cluster
B. Get hands-on experience with word count map reduce programs

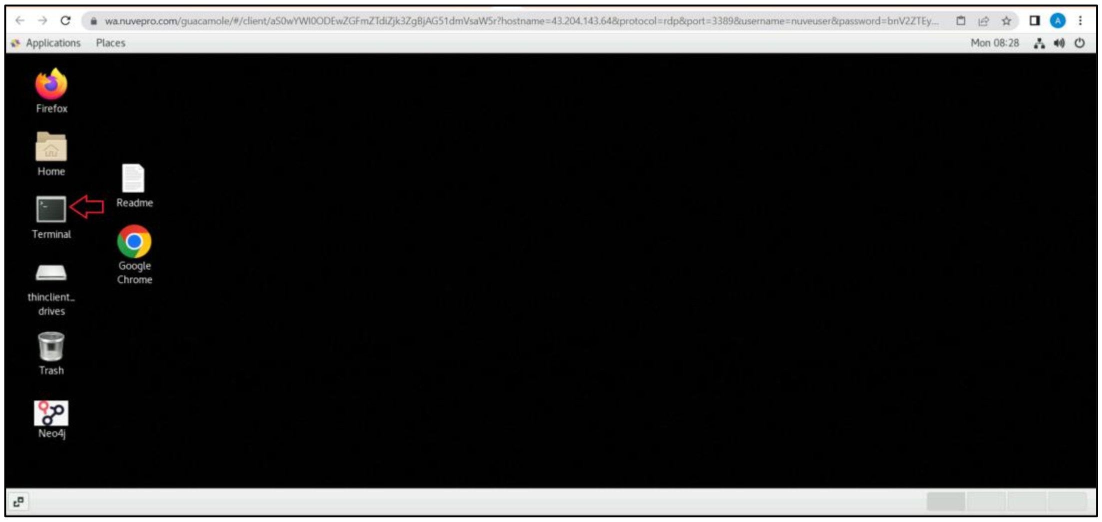
This lab sheet provides a quick introduction of using Spark for Map Reduce program with Python. This exercise will introduce the API through pySpark package, then next labs will show how to write applications in Python.

2. Steps to be performed

Note - It's assumed that student has made a slot reservation using the slot booking interface where Apache Spark framework was selected. The details of the Apache Spark systems to be used is received through an email. If not, please contact the administrators for the same.
Also it's assumed that students are aware of the process of logging into these virtual machines. If not, then get access to the user manual maintained for the usage of remote lab setup.

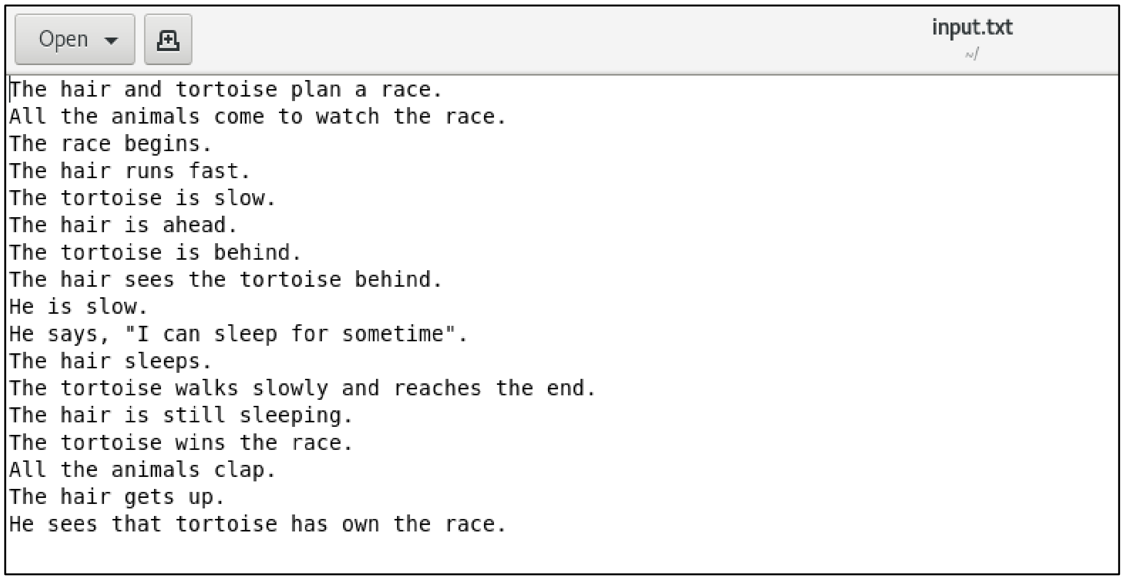
Preparations -
Data Preparations -

a) Open the terminal by clicking on icon on desktop



b) Prepare the input text file using any file editor. Copy and paste the content present in the attached input.txt file in this file.

```
[centos@master ~]$ gedit input.txt
```



```
The hair and tortoise plan a race.
All the animals come to watch the race.
The race begins.
The hair runs fast.
The tortoise is slow.
The hair is ahead.
The tortoise is behind.
The hair sees the tortoise behind.
He is slow.
He says, "I can sleep for sometime".
The hair sleeps.
The tortoise walks slowly and reaches the end.
The hair is still sleeping.
The tortoise wins the race.
All the animals clap.
The hair gets up.
He sees that tortoise has own the race.
```

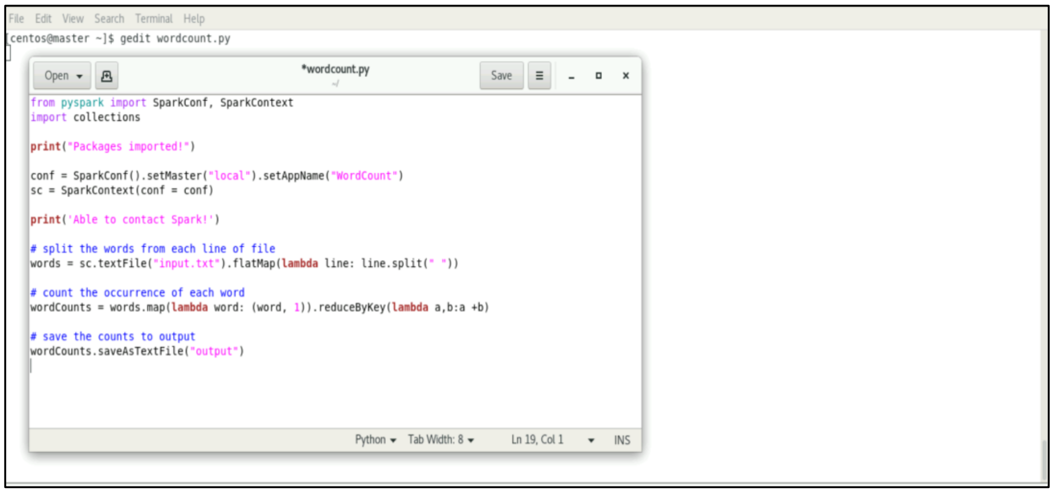
Installing pySpark

c) For the execution of python programmes on the Spark, a package named pyspark is required. Using the sudo previleges, install the packages with pip command.

pip install pyspark

Writing WordCount program

d) Open up the text editor and copy the code written in the attached wordcount.py file.



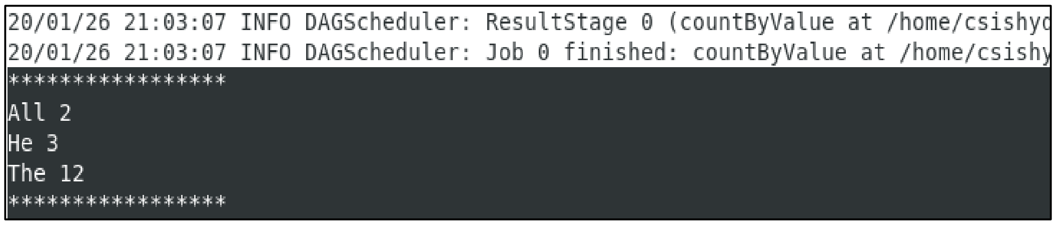
```
[centos@master ~]$ gedit wordcount.py
```

```
from pyspark import SparkConf, SparkContext
import collections

print("Packages imported!")

conf = SparkConf().setMaster("local").setAppName("WordCount")
sc = SparkContext(conf = conf)

print('Able to contact Spark!')

# split the words from each line of file
words = sc.textFile("input.txt").flatMap(lambda line: line.split(" "))

# count the occurrence of each word
wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)

# save the counts to output
wordCounts.saveAsTextFile("output")
```

e) Execute the wordcount.py file using the spark-submit command.

```
[centos@master ~]$ spark-submit wordcount.py
```

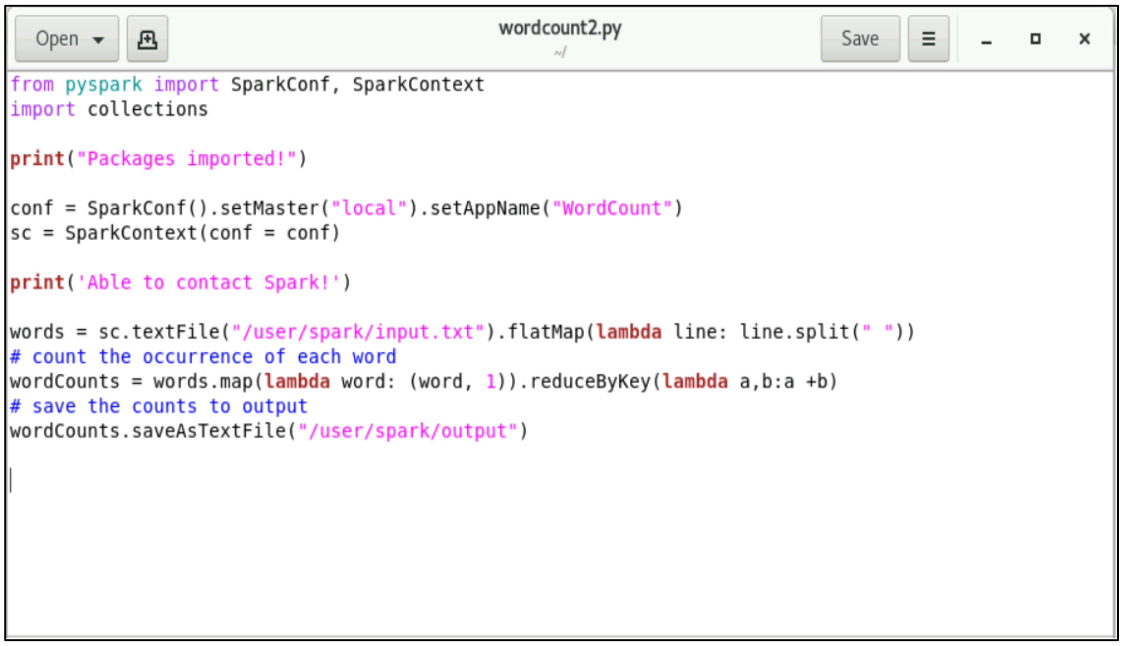
f) Look at the outcome printed while the program is getting executed on the Spark cluster. It shows how many times the first word of each lines has appeared.



```
20/01/26 21:03:07 INFO DAGScheduler: ResultStage 0 (countByValue at /home/csishyd
20/01/26 21:03:07 INFO DAGScheduler: Job 0 finished: countByValue at /home/csishy
*****************
All 2
He 3
The 12
*****************
```

g) Open up the text editor and copy the code written in the attached wordcount2.py file.

```
[centos@master ~]$ gedit wordcount2.py
```

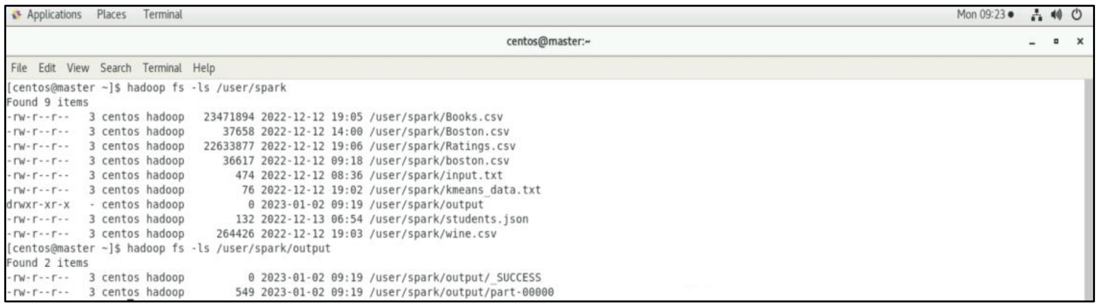


```python
from pyspark import SparkConf, SparkContext
import collections

print("Packages imported!")

conf = SparkConf().setMaster("local").setAppName("WordCount")
sc = SparkContext(conf = conf)

print('Able to contact Spark!')

words = sc.textFile("/user/spark/input.txt").flatMap(lambda line: line.split(" "))
# count the occurrence of each word
wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)
# save the counts to output
wordCounts.saveAsTextFile("/user/spark/output")
```

h) Execute the wordcount2.py file using the spark-submit command.

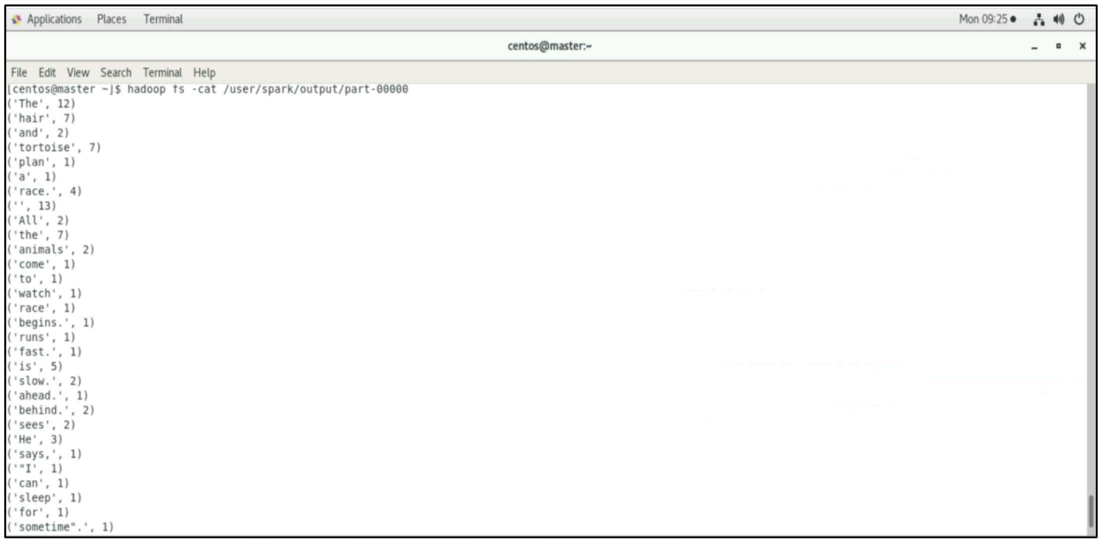
```
[centos@master ~]$ spark-submit wordcount2.py
```

i) Look at the outcome printed while the program is getting executed on the Spark cluster. It shows how many times the word of each lines has appeared. The output will be stored in the "output" directory as follows


Applications Places Terminal
Mon 09:23 ●
centos@master:~
File Edit View Search Terminal Help

```
[centos@master ~]$ hadoop fs -ls /user/spark
Found 9 items
-rw-r--r--   3 centos hadoop   23471894 2022-12-12 19:05 /user/spark/Books.csv
-rw-r--r--   3 centos hadoop      37658 2022-12-12 14:00 /user/spark/Boston.csv
-rw-r--r--   3 centos hadoop   22633877 2022-12-12 19:06 /user/spark/Ratings.csv
-rw-r--r--   3 centos hadoop      36617 2022-12-12 09:18 /user/spark/boston.csv
-rw-r--r--   3 centos hadoop        474 2022-12-12 08:36 /user/spark/input.txt
-rw-r--r--   3 centos hadoop         76 2022-12-12 19:02 /user/spark/kmeans_data.txt
drwxr-xr-x   - centos hadoop          0 2023-01-02 09:19 /user/spark/output
-rw-r--r--   3 centos hadoop        132 2022-12-13 06:54 /user/spark/students.json
-rw-r--r--   3 centos hadoop     264426 2022-12-12 19:03 /user/spark/wine.csv
[centos@master ~]$ hadoop fs -ls /user/spark/output
Found 2 items
-rw-r--r--   3 centos hadoop          0 2023-01-02 09:19 /user/spark/output/_SUCCESS
-rw-r--r--   3 centos hadoop        549 2023-01-02 09:19 /user/spark/output/part-00000
```

j) Look at the output in the file.


Applications Places Terminal
Mon 09:25 ●
centos@master:~
File Edit View Search Terminal Help

```
[centos@master ~]$ hadoop fs -cat /user/spark/output/part-00000
('The', 12)
('hair', 7)
('and', 2)
('tortoise', 7)
('plan', 1)
('a', 1)
('race.', 4)
('', 13)
('All', 2)
('the', 7)
('animals', 2)
('come', 1)
('to', 1)
('watch', 1)
('race', 1)
('begins.', 1)
('runs', 1)
('fast.', 1)
('is', 5)
('slow.', 2)
('ahead.', 1)
('behind.', 2)
('sees', 2)
('He', 3)
('says,', 1)
('"I', 1)
('can', 1)
('sleep', 1)
('for', 1)
('sometime".', 1)
```

3. Outputs/Results

Students should be able to

- Execute the python map reduce program on Spark cluster
- See the word counts produced by the program for the first word of every line of a file

4. Observations

Students carefully needs to observe the profiling information generated as a result of executing a spark program.

5. References

A. Spark Documentation
B. pySpark API Guide