Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Big Data Systems – Spark Lab Sheet 3

Word Count with Spark

1. Objective:

Students should be able to

A. Get familiarity with the execution of Python programmes on the Spark cluster

B. Get hands-on experience with word count map reduce programme

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

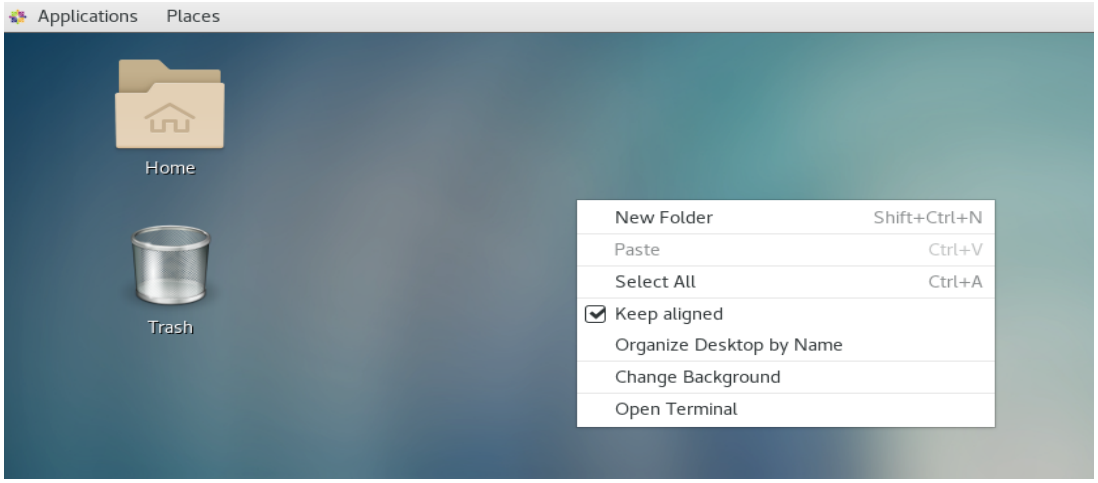
2. Steps to be performed:

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.

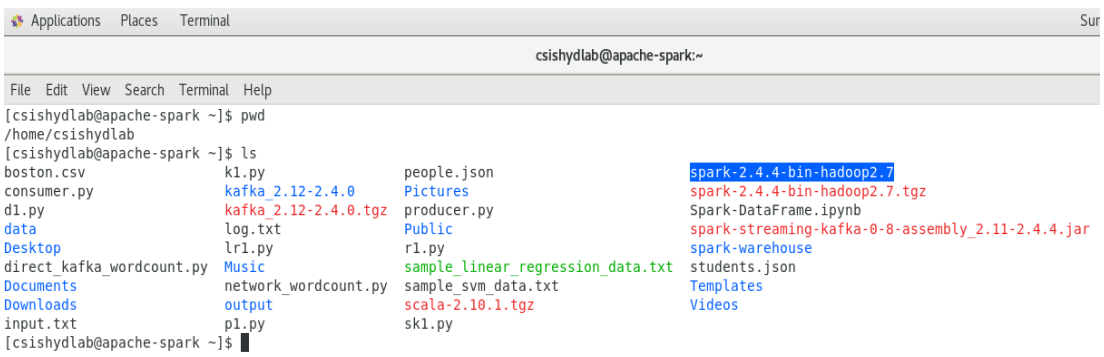
Also it's assumed that students are aware of the process of logging into these virtual machines. If not, then get access to the user manual maintained for the usage of remote lab setup.

Preparations -

a) Open the terminal by right clicking on the desktop of the virtual machine.



b) Look at the current directory and also file listings in it. It must have a spark installation directory present in it. Commands like pwd, ls can be used for it.



c) Set the SPARK_HOME and HOME variable to point to the spark installations.

[csishydlab@apache-spark bin]$ pwd

/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin

[csishydlab@apache-spark bin]$ export SPARK_HOME=/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin

[csishydlab@apache-spark bin]$ export PATH="$SPARK_HOME/bin:$PATH"

echo $SPARK_HOME

echo $PATH

```
Applications  Places  Terminal                                                    Sun 19:30
                                   csishydlab@apache-spark:~
File  Edit  View  Search  Terminal  Help
[csishydlab@apache-spark ~]$ pwd
/home/csishydlab
[csishydlab@apache-spark ~]$ ls
boston.csv                k1.py                  people.json                        spark-2.4.4-bin-hadoop2.7
consumer.py               kafka_2.12-2.4.0       Pictures                           spark-2.4.4-bin-hadoop2.7.tgz
d1.py                     kafka_2.12-2.4.0.tgz   producer.py                        Spark-DataFrame.ipynb
data                      log.txt                Public                             spark-streaming-kafka-0-8-assembly_2.11-2.4.4.jar
Desktop                   lr1.py                 r1.py                              spark-warehouse
direct_kafka_wordcount.py Music                  sample_linear_regression_data.txt  students.json
Documents                 network_wordcount.py   sample_svm_data.txt                Templates
Downloads                 output                 scala-2.10.1.tgz                   Videos
input.txt                 p1.py                  sk1.py
[csishydlab@apache-spark ~]$ export SPARK_HOME=/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin
[csishydlab@apache-spark ~]$ export PATH="$SPARK_HOME/bin:$PATH"
[csishydlab@apache-spark ~]$ echo $SPARK_HOME
/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin
[csishydlab@apache-spark ~]$ echo $PATH
/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/home/csishydlab/.local/bin:/home/cs
shydlab/bin:/usr/lib/scala/bin:/home/csishydlab/spark-2.4.4-bin-hadoop2.7/bin
```

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

```
File  Edit  View  Search  Terminal  Help
[csishydlab@apache-spark ~]$ gedit input.txt&
[1] 13546
[csishydlab@apache-spark ~]$
```

Installing pySpark

e) 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 programme

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

```
File  Edit  View  Search  Terminal  Help
[root@apache-spark csishydlab]# gedit wordcount.py &
[1] 20300
[root@apache-spark csishydlab]#
```

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

```
File  Edit  View  Search  Terminal  Help

[root@apache-spark csishydlab]# gedit wordcount.py &
[1] 20300
[root@apache-spark csishydlab]# spark-submit wordcount.py
20/01/26 21:03:04 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
Packages imported!
```

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

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

```
File  Edit  View  Search  Terminal  Help
[root@apache-spark csishydlab]# gedit wordcount2.py &
[1] 21632
[root@apache-spark csishydlab]#
```

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

```
csishydlab@apache-spark:/home/csishydlab
File  Edit  View  Search  Terminal  Help
[root@apache-spark csishydlab]# spark-submit wordcount2.py
```

k) 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

```
csishydlab@apach
File  Edit  View  Search  Terminal  Help
[root@apache-spark csishydlab]# ls
boston.csv                  input.txt              output
consumer.py                 k1.py                  p1.py
d1.py                       kafka_2.12-2.4.0       people.json
data                        kafka_2.12-2.4.0.tgz   Pictures
Desktop                     log.txt                producer.py
direct_kafka_wordcount.py   lr1.py                 Public
Documents                   Music                  r1.py
Downloads                   network_wordcount.py   sample_linear_regressior
[root@apache-spark csishydlab]# ls output/
part-00000  _SUCCESS
[root@apache-spark csishydlab]# gedit output/part-00000 &
```

l) Look at the output in the file.

Open

part-00000
/home/csishydlab/output

input.txt wordcount.py

```
(u'and', 2)
(u'', 13)
(u'own', 1)
(u'sees', 2)
(u'is', 5)
(u'reaches', 1)
(u'hair', 7)
(u'sleep', 1)
(u'walks', 1)
(u'fast.', 1)
(u'still', 1)
(u'slow.', 2)
(u'end.', 1)
(u'up.', 1)
(u'begins.', 1)
(u'for', 1)
(u'says.', 1)
```

3. Outputs/Results:

Students should be able to

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

4. Observations:

Students carefully needs to observe

- Details provided while spark application was running
- Number of maps executed
- Number of reducers used

5. References:

A. Spark Documentation

B. pySpark API Guide