ASSIGNMENT E1

Spark Example for CS4B

Submitted By: Ahmed Dider Rahat
Matriculation Number: 916146

Assignment Code: After implementing all the assignment I pushed my code on my Github link.

Answer to the question no. 1

I have completed the installation process by using some steps. They are:

1. **Java Installation:** I had already installed java 17 in my system. So, at first I remove my version of java and then install java 8 from https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html . There are several version of jdk uploaded. I choose the marked one.

Solaris SPARC 64-bit (SVR4 package)	133.66 MB	jdk-8u301-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit Compressed Archive	94.8 MB	jdk-8u301-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.42 MB	jdk-8u301-solaris-x64.tar.Z
Solaris x64 Compressed Archive	92.66 MB	jdk-8u301-solaris-x64.tar.gz
Windows x86 Installer	156.45 MB	jdk-8u301-windows-i586.exe
Windows x64 Installer	169.46 MB	jdk-8u301-windows-x64.exe

So, the current java version become:

```
C:\Users\DELL>java -version
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 1.8.0_301-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.301-b09, mixed mode)
```

2. **Python Installation:** Python 3 already installed before. So, just check the version:

```
C:\Users\DELL>python --version
Python 3.6.5 :: Anaconda, Inc.
```

3. **Folder Creation:** Create two empty folder in C drive. One is Spark and another one is Hadoop/bin.

4. **Download Spark:** Download spark from https://spark.apache.org/downloads.html and download the spark zip file.

Download Apache Spark™

1. Choose a Spark release: 3.0.3 (Jun 23 2021)
2. Choose a package type: Pre-built for Apache Hadoop 2.7
3. Download Spark: spark-3.0.3-bin-hadoop2.7.tgz
4. Verify this release using the 3.0.3 signatures, checksums and project release KEYS.

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

After download the zip file, I unzip it and move it to the C:\Spark location.

5. **Download winutils:** I download the winutils file from github. As I used hadoop 2.7 my link was https://github.com/cdarlint/winutils/tree/master/hadoop-2.7.7/bin. Then I move it to C:\hadoop\bin.

6. Setup environment variables: For the setting I need to setup the environment variables. So, I added them in the environment variable.

User variables for DELL

Variable	Value
HADOOP_HOME	C:\hadoop
IntelliJ IDEA Community E...	C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.2\bin;
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_301
JAVA_PATH	C:\Program Files\Java\jdk1.8.0_301\bin
MALLET_HOME	C:\mallet-2.0.8
OneDrive	C:\Users\DELL\OneDrive
OneDriveConsumer	C:\Users\DELL\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\DELL\AppData\Local\Programs\Py
PyCharm Community Editi...	C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\bin;
SPARK_HOME	C:\Spark\spark-3.0.3-bin-hadoop2.7
TEMP	C:\Users\DELL\AppData\Local\Temp

Then add the path of spark and hadoop in the path section.

Edit environment variable

C:\Program Files\MySQL\MySQL Shell 8.0\bin\
C:\Users\DELL\AppData\Local\Programs\Python\Python36\Scrip...
C:\Users\DELL\AppData\Local\Programs\Python\Python36\
C:\Users\DELL\AppData\Local\Programs\Python\Launcher\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\CodeBlocks\MinGW\bin
%IntelliJ IDEA Community Edition%
%PyCharm Community Edition%
%SPARK_HOME%\bin
%HADOOP_HOME%\bin

Answer to the question no. 2

After successfully installation my Anaconda promt become:

```
Anaconda Prompt - pyspark

(base) C:\Users\DELL>pyspark
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
22/01/23 18:45:48 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.3
      /_/

Using Python version 3.6.5 (default, Mar 29 2018 13:32:41)
SparkSession available as 'spark'.
```

i) **Run simple python command and see the output:**

```
Anaconda Prompt - pyspark
>>> list = [1, 22, 2, 3, 4]
>>> print(sorted(list))
[1, 2, 3, 4, 22]
```

ii) **Run pi.py:**

```
C:\Spark\spark-3.0.3-bin-hadoop2.7\examples\src\main\python>pyspark < pi.py
22/01/23 19:28:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Pi is roughly 3.141040

C:\Spark\spark-3.0.3-bin-hadoop2.7\examples\src\main\python>SUCCESS: The process with PID 20168 (child process of PID 18
392) has been terminated.
SUCCESS: The process with PID 18392 (child process of PID 14916) has been terminated.
SUCCESS: The process with PID 14916 (child process of PID 21900) has been terminated.
```

iii) **spark-submit --master local[4] pi.py 2>session.log:**

```
C:\Spark\spark-3.0.3-bin-hadoop2.7\examples\src\main\python>spark-submit --master local[4] pi.py 2>session.log
Pi is roughly 3.146460

C:\Spark\spark-3.0.3-bin-hadoop2.7\examples\src\main\python>
```

Answer to the question no. 3

In [2]:
```
from __future__ import print_function

import sys
from random import random
from operator import add

from pyspark.sql import SparkSession
```

In [3]:
```
if __name__ == "__main__":

    spark = SparkSession\
        .builder\
        .appName("PythonPi")\
        .getOrCreate()

    slices = 1 #int(sys.argv[1]) if len(sys.argv) > 1 else 2
    n = 100000 * slices

    def f(_):
        x = random() * 2 - 1
        y = random() * 2 - 1
        return 1 if x ** 2 + y ** 2 <= 1 else 0

    count = spark.sparkContext.parallelize(range(1, n + 1), slices).map(f).reduce(add)
    print("Pi is roughly %f" % (4.0 * count / n))

    spark.stop()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pyspark\context.py:238: FutureWarning: Python 3.6 support is deprecated in Spark 3.
2.
  FutureWarning
```

```
Pi is roughly 3.144640
```