Lab: Analyzing Data with Spark
Cloud Computing For Data Analysis

You can read about Spark in Chapter 19 of the Hadoop book (4th ed) by White, and in the book Learning Spark by Karau, Konwinski, Wendell, and Zaharia.

We will use the Python interface to Spark in this document.

Replace <username> with your 49er Username throughout the document.

Running on DSBA cluster:

First please ensure you have access to the dsba-hadoop cluster.
To access the cluster from a Unix machine, type:

```
$ ssh -X dsba-hadoop.uncc.edu -l <username>
```

Password is your 49er password.

Note: If you don't have access, send email to the instructor and the TA.

Step 1: Startup Spark using its Python interface:

Copy an input file for next steps over to hdfs. For example: at regular shell:

```
$ hadoop fs -put WordCount.java /user/<username>/.
```

```
$ pyspark
```

A Spark program first creates a SparkContext object, which tells Spark how to access a cluster. This is the sc variable, which is created automatically. Other programs must use a constructor to instantiate a new SparkContext.

```
>>> sc
```

Creating an RDD (resilient distributed dataset)

```
>>> data= [1, 2, 3, 4, 5, 6, 7, 8] # a list

>>> data

>>> rddData = sc.parallelize(data) # Create an RDD called rddData

>>> rddData
```

Spark can also create RDDs from any file stored in HDFS or other storage systems supported by Hadoop. Can also take a directory as input.

```
>>> lines = sc.textFile("WordCount.java") # Create an RDD called lines
>>> lines.count() # an action to count the number of items in the RDD
>>> lines.first() # another action to get the first item in the RDD

>>> os.system('ls')  # you can execute system commands from the python shell
```

[To exit shell, press Ctrl-D]

Step 2: Passing functions to Spark operators:

2.1 Closures with lambda expressions:

```
>>> WordLines = lines.filter(lambda line: "Word" in line) # transformed RDD
>>> WordLines.count()
>>> WordLines.first()
```

2.2 Defining a named function:

```
>>> def hasText(line):
>>> return "Text" in line # make sure to add <Space> before “return” and then press <Enter> twice.

>>> TextLines = lines.filter(hasText)
>>> TextLines.count()
>>> TextLines.first()
```

Step 3: Transformations and Actions

There are two types of operations on RDDs in Spark:

1. Transformations
2. Actions

3.1 : Transformations

Example transformations include map(), flatMap(), filter().

```
>>> squareData = rddData.map(lambda x: x*x)

>>> oddData = rddData.filter(lambda x: x % 2 )

>>> oddSquareData = oddData.map(lambda x: x*x)

>>> osd = oddSquareData.collect()

>>> osd
```

Comparison of map() and flatMap().

```
>>> distFile =sc.textFile("WordCount.java")

>>> dfm= distFile.map(lambda x: x.split(' ')).collect()

>>> dfm # map returns only one element a list of strings per line

>>> dffm = distFile.flatMap(lambda x: x.split(' ')).collect()

>>> dffm # flatMap returns a list of 0 or more elements as an iterator per line and its output is
flattened
```

We can concatenate transformations and actions.

```
>>> oddSquareData2 = rddData.filter(lambda x: x % 2).map(lambda x: x*x)

>>> osd2 = oddSquareData2.collect()

>>> osd2
```

We can persist (i.e., cache) an RDD to avoid recomputing it repeatedly.

```
>>> oddData.persist()

>>> oddSquareData3 = oddData.map(lambda x: x*x)
```

3.2 : Actions

Example actions include reduce(), collect(), first(), count(), countByValue(), countByKey().

```
>>> WordLines.collect() # retrieves entire RDD

>>> data = [('hello',1),('world',1),('from',1),('hadoop',1),('world',1)]

>>> rddData = sc.parallelize(data)
>>> wordcounts = rddData.reduceByKey(lambda x,y: x+y) # sum up values of same key

>>> wordcounts.collect()

>>> wordgrps = rddData.groupByKey() # group up values of same key

>>> wordgrps.collect()
```

When RDDs are too large to be collect()ed to the memory of a single machine, we can write data out to storage using actions such as saveAsTextFile().

```
>>> distFile.flatMap(lambda x: x.split(' ')).saveAsTextFile("spark.flatMap.dump") # creates a directory
```

For more information about Spark programming see:
http://spark.apache.org/docs/latest/programming-guide.html

Exercise task: Implement WordCount in Spark.