CSP554—Big Data Technologies

Assignment #06 (Modules 06)

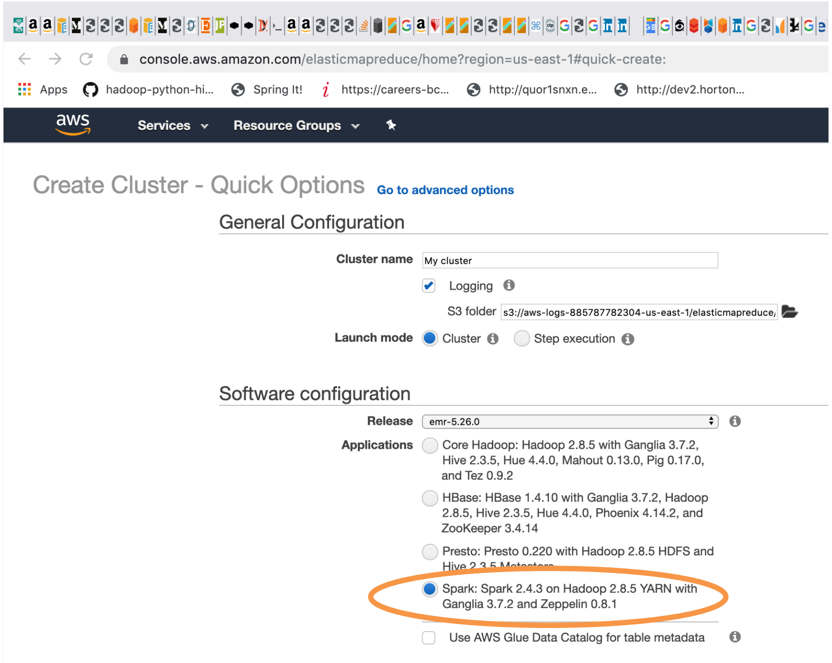
Worth: 12 points (2 points for each of the six problems)

Some basic notes:

- We will again be using files generated by the program TestDataGen. But even though the files this program generates end is the '.txt' suffix, I want you to treat them as if they were '.csv' files. In fact, if you like, when you copy them to HDFS you can change their suffixes form '.txt' to '.csv'. But this is not necessary to complete the exercises.
- Sometimes when specifying a filter or other condition below you need to keep operator precedence in mind. Simply stated, if you write something like "a == b & c < d" it may happen that this would be parsed as "(a == b & c) < d" which is not what you might expect or want. So, always use full parentheses to indicate your intent. For example, "((a == b) & (c < d))" will make sure that the condition is likely what you want.

Setting up to Use Spark in EMR:

Start up a Hadoop cluster as previously, but instead of choosing the "Core Hadoop" configuration chose the "Spark" configuration (see below), otherwise proceed as before.



console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Create Cluster - Quick Options

General Configuration

Cluster name: My cluster

☑ Logging

S3 folder: s3://aws-logs-885787782304-us-east-1/elasticmapreduce/

Launch mode: ● Cluster ○ Step execution

Software configuration

Release: emr-5.26.0

Applications:

○ Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.5, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2

○ HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.5, Hue 4.4.0, Phoenix 4.14.2, and ZooKeeper 3.4.14

○ Presto: Presto 0.220 with Hadoop 2.8.5 HDFS and Hive 2.3.5 Metastore

● Spark: Spark 2.4.3 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1

□ Use AWS Glue Data Catalog for table metadata

Exercises

Exercise 1)

Use the TestDataGen program from previous assignments to generate a new foodratings<magic_number>.txt data file.

Copy the file to HDFS, say into the /user/hadoop directory.

Read in the text file into an RDD named ex1RDD.

This RDD should now have records each consisting of a single string having 6 comma-separated parts something like the following:

u'Joe,44,33,41,1,5'

u'Mel,13,33,30,50,6'

u'Mel,12,40,30,42,1'

u'Sam,15,28,28,39,2'

List the first five records of the RDD using the “take(5)” action and copy them and the “magic number to your assignment submission for this exercise.

Exercise 2)

Create another RDD called ex2RDD where each record of this new RDD has 6 fields, each a string, by splitting apart each record on “,” boundaries from the ex1RDD.

The records of the new RDD should look something like:

u'Joe', u'44', u'33', u'41', u'1‘, u’5’

u‘Mel', u'13', u'33', u'30', u'50, u’6’‘

u‘Mel', u'12', u'40', u'30', u'42‘, u’1’

u'Sam', u'15', u'28', u'28', u'39‘, u’3’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

Exercise 3)

Create another RDD called ex3RDD from ex2RDD where each record of this new RDD has its third column converted from a string to an integer.

The records of the new RDD should look something like:

u'Joe', u'44', 33, u'41', u'1‘, u’1’

u‘Mel', u'13', 33, u'30', u'50‘, u’2’

u‘Mel', u'12', 40, u'30', u'42‘, u’3’

u'Sam', u'15', 28, u'28', u'39‘, u’4’

Hint: Use a lambda function something like the following:

```
lambda line : [line[0], line[1], int(line[2]), line[3], line[4], line[5]]
```

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

Exercise 4)

Create another RDD called ex4RDD from ex3RDD where each record of this new RDD is allowed to have a value for its third field that is less than 25 (<25).

The records of the new RDD should look something like:

u'Joe', u'44', 21, u'41', u'1‘, u’6’

u‘Mel', u'13', 3, u'30', u'50‘, u’1’

u‘Mel', u'12', 4, u'30', u'42‘, u’4’

u'Sam', u'15', 8, u'28', u'39‘, u’5’

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

Exercise 5)

Create another RDD called ex5RDD from ex4RDD where each record is a key value pair where the key is the first field of the record and the value is the entire record

The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’1’))

(u’Mel’, (u‘Mel', u'13', 3, u'30', u'50‘, u’6’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.

Exercise 6)

Create another RDD called ex6RDD from ex5RDD where the records are organized in ascending order by key

The records of the new RDD should look something like:

(u’Joe’, (u'Joe', u'44', 21, u'41', u'1‘, u’4’))

(u’Mel’ , (u‘Mel', u'13', 3, u'30', u'50‘, u’3’))

(u’Sam’ , (u‘Sam', u'23', 3, u'40', u'20‘, u’7’))

List the first five records of this RDD using the “take(5)” action and copy them to your assignment submission for this exercise.