Spark Developer Training - 3 Days

Manaranjan Pradhan
manaranjan@enablecloud.com

Lab: Twitter Trends using Spark Streaming

Import all spark streaming libraries

In [ ]:

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
```

Create a spark streaming context

- With a batch interval of 10 seconds. The messages would accumulate for 10 seconds and then get processed.
- The check point directory is going to save the messages to recover in case of streaming components fail.

In [ ]:

```
ssc = StreamingContext(sc, 10 )
ssc.checkpoint( "file:///home/hadoop/lab/programs/trends/checkpoint")
```

Connect to the broker to recieve the tweets

- This should be the IP address and port number of your windows or mac machine, where the program TweetRead.py is running

In [ ]:

```
lines = ssc.socketTextStream("192.168.0.139", 5555)
```

Tokenize the tweets and extracts only those words that start with # (tweet tags)

In [ ]:

```
tags = lines.flatMap( lambda line:
                     line.split(" ") ).filter( lambda word:
                                              str(word).startswith("#") )
```

The tokens should be counted for a 20 seconds window and slided by 10 seconds

In [ ]:

```
tag_60 = tags.map( lambda tag: (tag, 1) ).reduceByKeyAndWindow( lambda a, b:
                                                               a + b, 20, 10 )
```

In [ ]:

```
#tag_60 = tags.map( lambda tag: (tag, 1) ).reduceByKey( lambda a, b: a + b )
```

Sort the tags by their counts

In [ ]:

```
top_tag_60 = tag_60.repartition(1).transform( lambda rdd:
                                             rdd.sortBy( lambda rec:
                                                        rec[1], ascending = False ) )
```

Define a function to write the results to a file

In [ ]:

```
import time
```

In [ ]:

```
def writeTweetTags(eachpart):
    with open('/home/hadoop/lab/programs/trends/results', 'a') as ftag:
        ftag.write('\n\n\n\n\n')
        ftag.write('############################################################')
        ftag.write('\n')
        ftag.write("Trends at: ")
        ftag.write(time.strftime("%H:%M:%S"))
        ftag.write('\n')
        ftag.write('-----------------------------------------')
        ftag.write('\n')
        for eachrecord in eachpart:
            ftag.write(str( eachrecord[0] ) + '\t' + str( eachrecord[1] ) + '\n' )
        ftag.write('############################################################')
```

Finally for each results partition, call the function writeTweetTags() to write to the files

In [ ]:

```
top_tag_60.foreachRDD(lambda tweetTagPart:
                      tweetTagPart.foreachPartition( lambda tweetTag:
                                                    writeTweetTags( tweetTag ) ) )
```

Start the streaming Process

In [ ]:

```
ssc.start()             # Start the computation
ssc.awaitTermination()  # Wait for the computation to terminate
```

Finally stop the streaming context

In [ ]:

```
ssc.stop()
```