Spark Streaming installation and execution:

Items needed for Spark Streaming:

SparkContext

StreamingContext

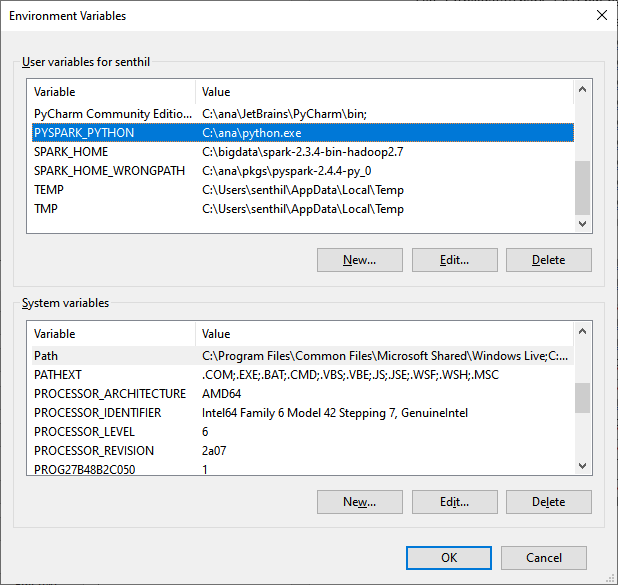
Code: SparkStreaming.py

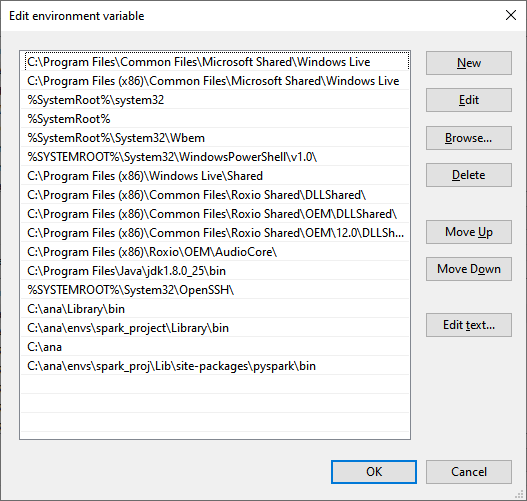
from pyspark import SparkContext  
from pyspark.streaming import StreamingContext  
# Creating a local StreamingContext with three working thread and batch intervel of 1 sec  
sc = SparkContext("local[3]",appName="NetworkWordCount")  
  
ssc = StreamingContext(sc,1)  
  
#create a DStream that will connect to hostname:port , such as localhost:9999  
lines = ssc.socketTextStream("localhost",9999)  
  
  
#This lines DStream represents the stream of data that will be received from the data server. Each record in this DStream is a line of text. Next, we want to split the lines by space into words.  
#split each line into words  
words = lines.flatMap(lambda line: line.split(" "))  
  
  
#flatMap is a one-to-many DStream operation that creates a new DStream by generating multiple new records from each record in the source DStream. In this case, each line will be split into multiple words and the stream of words is represented as the words DStream. Next, we want to count these words.  
  
#count each word in each batch  
pairs = words.map(lambda word: (word,1))  
wordCounts = pairs.reduceByKey(lambda x, y: x+y)  
  
# Print the first ten elements of each RDD generated in this DStream to the console  
wordCounts.pprint()  
  
  
#The words DStream is further mapped (one-to-one transformation) to a DStream of (word, 1) pairs, which is then reduced to get the frequency of words in each batch of data. Finally, wordCounts.pprint() will print a few of the counts generated every second.  
  
#Note that when these lines are executed, Spark Streaming only sets up the computation it will perform when it is started, and no real processing has started yet. To start the processing after all the transformations have been setup, we finally call  
  
ssc.start() # Start the computation  
  
ssc.awaitTermination() # Wait for the computation to terminate  
'''  
'''

|  |  |  |
| --- | --- | --- |
| Spark execution path | spark-submit c:\ana\pycharmprojects\clo\spark\_proj\sparkstreaming.py |  |

Python Spark – Path setup

PYSPARK\_PYTHON = C:\ANA\python.exe





Error:

Spark Streaming not working in 2.4 version for SPARK. Need to install Spark 2.3 version and set the SPARK\_HOME = C:\bigdata\spark-2.3.4-bin-hadoop2.7

|  |
| --- |
| File "C:\bigdata\spark-2.4.0-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\worker.py", line 25, in <module> |
| ModuleNotFoundError: No module named 'resource' |
| 2019-10-17 09:49:06 INFO JobScheduler:54 - Added jobs for time 1571323746000 ms |
| 2019-10-17 09:49:07 INFO JobScheduler:54 - Added jobs for time 1571323747000 ms |
| 2019-10-17 09:49:08 INFO JobScheduler:54 - Added jobs for time 1571323748000 ms |
| 2019-10-17 09:49:09 INFO JobScheduler:54 - Added jobs for time 1571323749000 ms |
| 2019-10-17 09:49:10 INFO JobScheduler:54 - Added jobs for time 1571323750000 ms |
| 2019-10-17 09:49:11 INFO JobScheduler:54 - Added jobs for time 1571323751000 ms |
| 2019-10-17 09:49:12 INFO JobScheduler:54 - Added jobs for time 1571323752000 ms |
| 2019-10-17 09:49:13 INFO JobScheduler:54 - Added jobs for time 1571323753000 ms |
| 2019-10-17 09:49:14 INFO JobScheduler:54 - Added jobs for time 1571323754000 ms |
| 2019-10-17 09:49:14 ERROR Executor:91 - Exception in task 0.0 in stage 2.0 (TID 1) |
| org.apache.spark.SparkException: Python worker failed to connect back. |
| at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker(PythonWorkerFactory.scala:170) |
| at org.apache.spark.api.python.PythonWorkerFactory.create(PythonWorkerFactory.scala:97) |
| at org.apache.spark.SparkEnv.createPythonWorker(SparkEnv.scala:117) |
| at org.apache.spark.api.python.BasePythonRunner.compute(PythonRunner.scala:108) |
| at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:65) |
| at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324) |
| at org.apache.spark.rdd.RDD.iterator(RDD.scala:288) |
| at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:65) |
| at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324) |
| at org.apache.spark.rdd.RDD.iterator(RDD.scala:288) |
| at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90) |
| at org.apache.spark.scheduler.Task.run(Task.scala:121) |
| at org.apache.spark.executor.Executor$TaskRunner$$anonfun$10.apply(Executor.scala:402)  NetCAT to test TCP data back and forth:  <https://joncraton.org/blog/46/netcat-for-windows/>  <https://joncraton.org/files/nc111nt.zip>  Pwd:nc  Download Nc111nt.zip  And unzip and store it a folder  Local machine  C:>nc111nt> nc -l -p 9999  Remote machine  spark-submit c:\ana\pycharmprojects\clo\spark\_proj\networdcount.py localhost 9999  When you type any word on local machine, Remote machine get the string and count the words. |