package streamTCP

import org.apache.spark.\_

import org.apache.spark.streaming.StreamingContext

import org.apache.spark.storage.StorageLevel

import org.apache.spark.streaming.\_

import org.apache.spark.streaming

object SparkStreaming {

def main(args : Array[String]) : Unit = {

//set the configuration for the spark application

val conf = new SparkConf().setAppName("Spark Streaming with TCP socket").setMaster("local[2]")

//create instance of SparkContext

val sc = new SparkContext(conf)

//set the batch interval

val batchInterval =10

//create instance of StreamingContext, Spark streaming runs on top of SparkContext

val ssc = new StreamingContext(sc,Seconds(batchInterval))

//Create the DStream data from the TCP socket stream source

val data = ssc.socketTextStream("localhost",9999, StorageLevel.MEMORY\_AND\_DISK\_SER)

//split the linee with delimiter " "

val words = data.flatMap(lines => lines.split(" "))

//map the words by 1

val length = words.map(words => (words,1))

//count the values associated with the words to get the count

val countWords = length.reduceByKey(\_+\_)

//print the Dstream

countWords.print()

//start the streamingContext

ssc.start()

//Makes the application thread wait for the stream computation to stop

ssc.awaitTermination()

}

}

