Once you have made your program more resilient to the input, run it on the whole dataset, and answer the following questions:

"""Lab 3. Basic wordcount

"""

from mrjob.job import MRJob

import re

import time

#this is a regular expression that finds all the words inside a String

WORD\_REGEX = re.compile(r"\b\w+\b")

#This line declares the class Lab1, that extends the MRJob format.

class Lab3(MRJob):

# this class will define two additional methods: the mapper method goes here

def mapper(self, \_, line):

fields = line.split(";")

#Fields contains line as follows.

# 0 1 2 3

#epoch\_time ; tweetId ; tweet(including #hashtags) ; device

#in order to select the tweet, you would choose fields[2].

try:

if (len(fields)==4):

#access the fields you want, assuming the format is correct now

tweet = fields[2]

time\_epoch = int(fields[0])/1000

day = time.strftime("%d",time.gmtime(time\_epoch)) #returns day of the month

#and the reducer method goes after this line

yield(day, fields)

except:

pass

#no need to do anything, just ignore the line, as it was malformed

def combiner (self, day, fields): # Reducer input (key, list of values)

yield(day,1)

def reducer(self, day, counts ):

yield (day, sum(counts))

#you have to implement the body of this method. Python's sum() function will probably be useful

#this part of the python script tells to actually run the defined MapReduce job. Note that Lab1 is the name of the class

if \_\_name\_\_ == '\_\_main\_\_':

# Lab2.JOBCONF= { 'mapreduce.job.reduces': '1' }

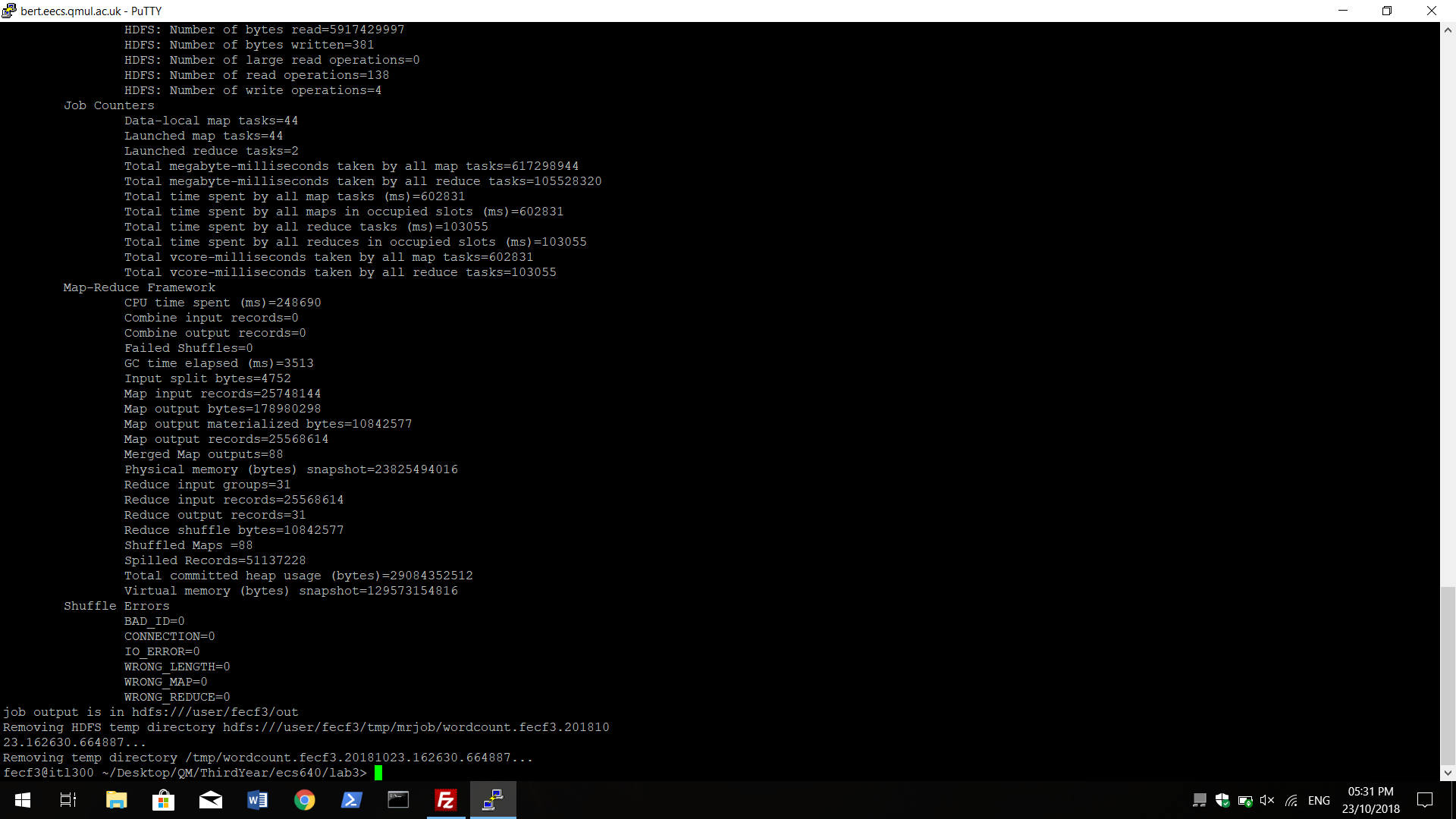
Lab3.run()

#python wordcount.py sherlock.txt > out.txt (as sherlock is the same folder location)

1. **What is the most popular day regarding twitter activity? If you do a quick online search you will probably be able to explain why.**

On saturday 6th August→ Fireworks exploded during the Opening Ceremony of the Rio 2016

1. **How many lines of input were malformed? You can figure out the answer by looking at the result**

****

Total malformed lines = Mapper inputs - MApper outputs

= 25748144 - 25568614

**= 179530**

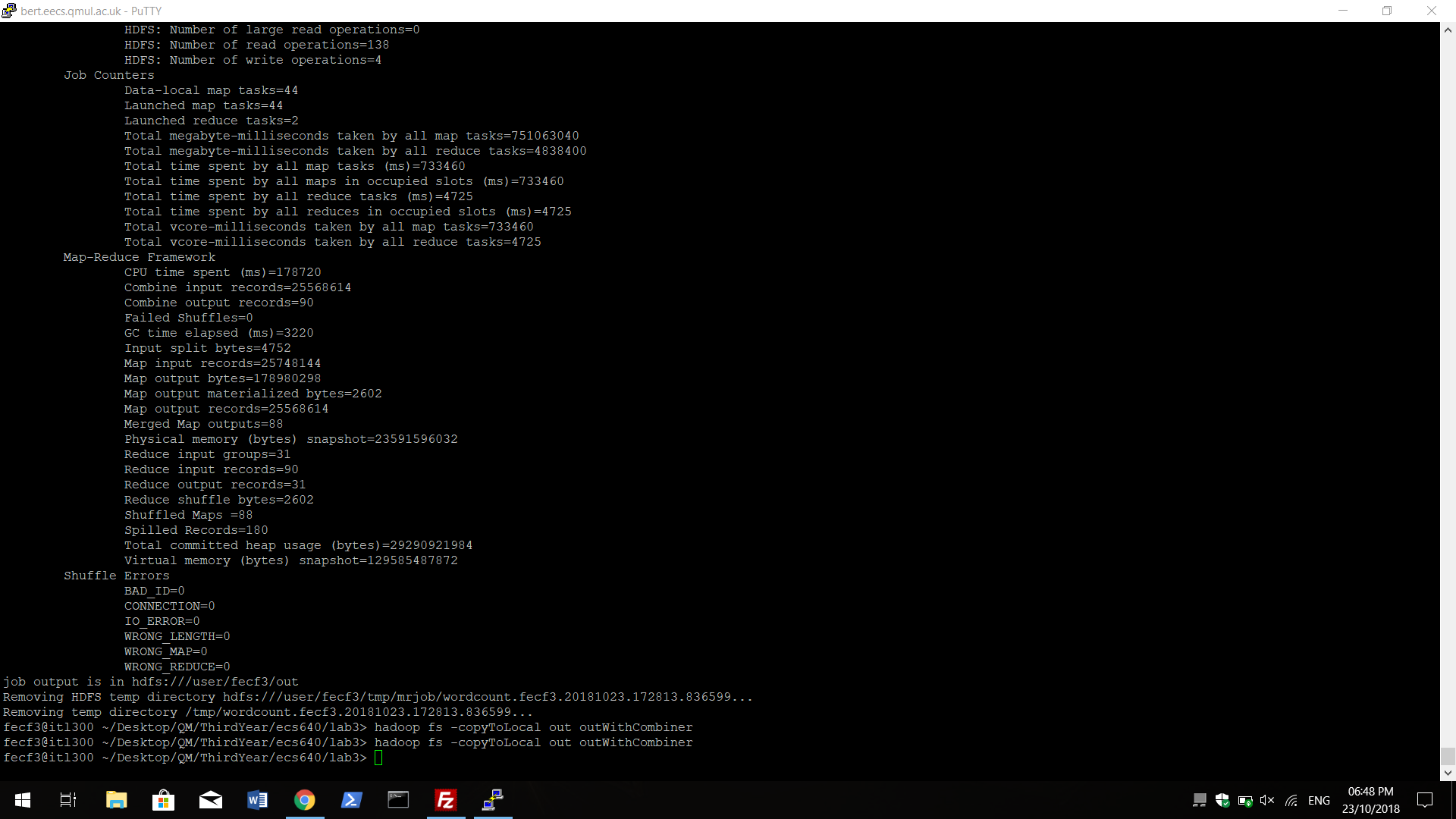
1. **Would this work benefit from a combiner? If so, add a working combiner and rerun the code. Compare the runtime statistics of both jobs, and try to estimate what is the effective impact of running the job with a combiner.**

**Runtime (no combiner) - CPU time spent (ms):** 248690

**Runtime (with combiner) - CPU time spent (ms):** 178720

Effective impact of running the job with the combiner: 248690/178720 = 1.3915

This means that the code with the combiner runs 1.39 times faster than the one without the combiner.



**4. What is the average number of hashtags? What is the average length of these tweets?**

"""Lab 3 Basic wordcount

"""

from mrjob.job import MRJob

import re

import time

#this is a regular expression that finds all the words inside a String

WORD\_REGEX = re.compile(r"\b\w+\b")

#This line declares the class Lab1, that extends the MRJob format.

class Lab3(MRJob):

# this class will define two additional methods: the mapper method goes here

def mapper(self, \_, line):

fields = line.split(";")

#Fields contains line as follows.

# 0 1 2 3

#epoch\_time ; tweetId ; tweet(including #hashtags) ; device

#in order to select the tweet, you would choose fields[2].

lines = 0

try:

if (len(fields)==4):

#access the fields you want, assuming the format is correct now

tweet = fields[2]

tweetLength = len(tweet)

ht = tweet.count('#')

# time\_epoch = int(fields[0])/1000

# day = time.strftime("%d",time.gmtime(time\_epoch)) #returns day of the month

#and the reducer method goes after this line

yield("Total length", tweetLength)

yield("Total Hashtag", ht)

except:

pass

#no need to do anything, just ignore the line, as it was malformed

def reducer(self, tweet, sums):

total = 0

count = 0

for i in sums:

total = total + i

count = count + 1

avg = total/count

yield(tweet,avg)

# def combiner (self, tweet, counts): # Reducer input (key, list of values)

# yield(tweet,sum(counts))

#you have to implement the body of this method. Python's sum() function will probably be useful

#this part of the python script tells to actually run the defined MapReduce job. Note that Lab1 is the name of the class

if \_\_name\_\_ == '\_\_main\_\_':

# Lab2.JOBCONF= { 'mapreduce.job.reduces': '1' }

Lab3.run()

#python wordcount.py sherlock.txt > out.txt (as sherlock is the same folder location)

**"Total Hashtag" 2.0587802341689887**

**"Total length" 109.13982740333867**

**To find the numbe4 of replicas: Asnwer: 10**

**bash-4.2$ hdfs dfs -stat %r /data/olympictweets2016rio^C**