ASSIGNMENT - 3

**CSP - 554 BIG DATA TECHNOLOGIES**

**Question 6**

Submit a copy of this modified program and a screen shot of the results of the program’s

execution as the output of your assignment.

**Program:**

**WordCount2.py**

from mrjob.job import MRJob

import re

WORD\_RE = re.compile(r&quot;[\w&#39;]+&quot;)

class MRWordCount(MRJob):

def mapper(self, \_, line):

for word in WORD\_RE.findall(line):

if (word[0]&gt;=&quot;a&quot; and word[0]&lt;=&quot;n&quot;):

yield &quot;a\_to\_n&quot;, 1

else:

yield &quot;other&quot;, 1

def combiner(self, word, counts):

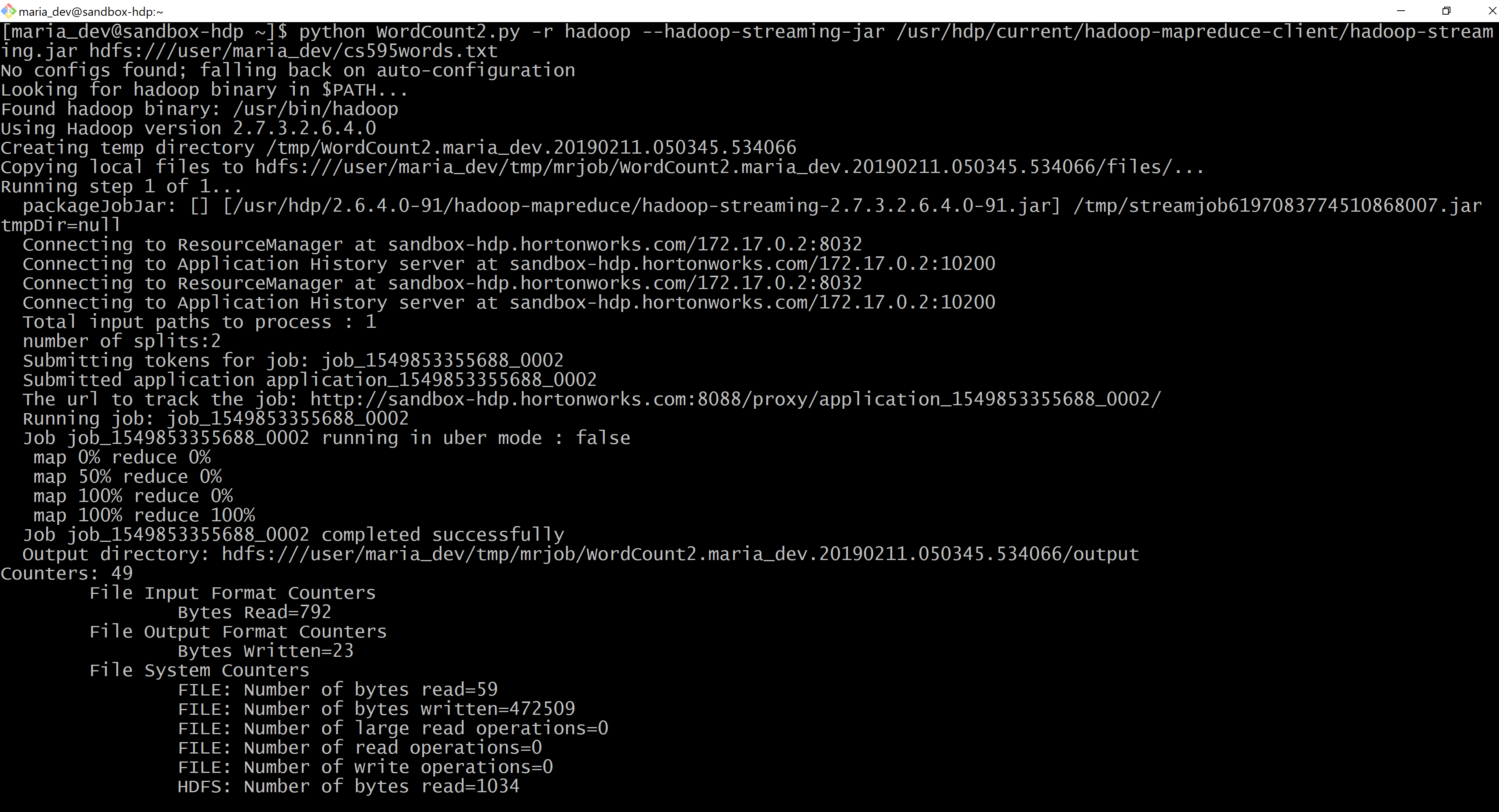
yield word, sum(counts)

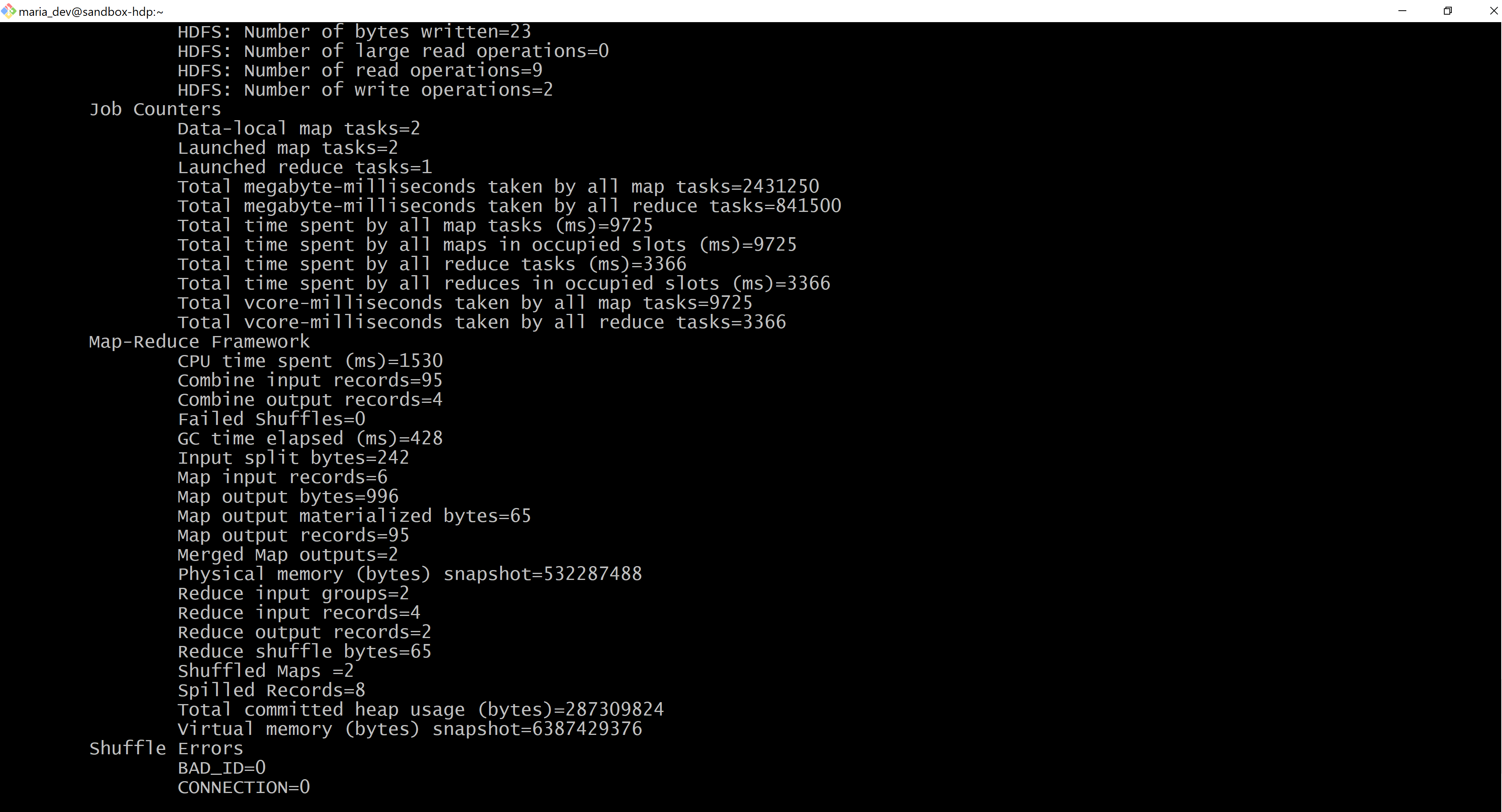
def reducer(self, word, counts):

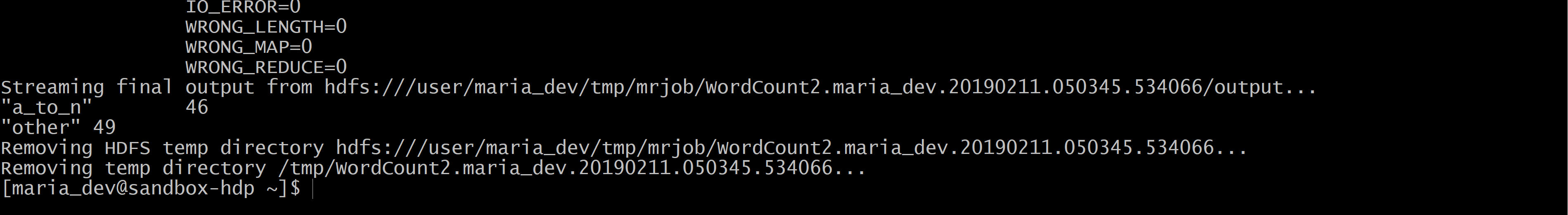
yield word, sum(counts)

if \_\_name\_\_ == &#39;\_\_main\_\_&#39;: MRWordCount.run()

**Output:**







**Question 10**

Submit a copy of this modified program and a screen shot of the results of the program’s execution as the output of your assignment.

**Program:**

**Salaries2.py**

from mrjob.job import MRJob

class MRSalaries(MRJob):

def mapper(self, \_, line):

(name,jobTitle,agencyID,agency,hireDate,annualSalary,grossPay) = line.split(&#39;\t&#39;)

if float(annualSalary)&gt;=0.0 and float(annualSalary)&lt;=49999.99:

yield &quot;Low&quot;, 1

elif float(annualSalary)&gt;=50000.0 and float(annualSalary)&lt;=99999.99:

yield &quot;Medium&quot;, 1

elif float(annualSalary)&gt;=100000.0:

yield &quot;High&quot;, 1

def combiner(self, jobTitle, counts):

yield jobTitle, sum(counts)

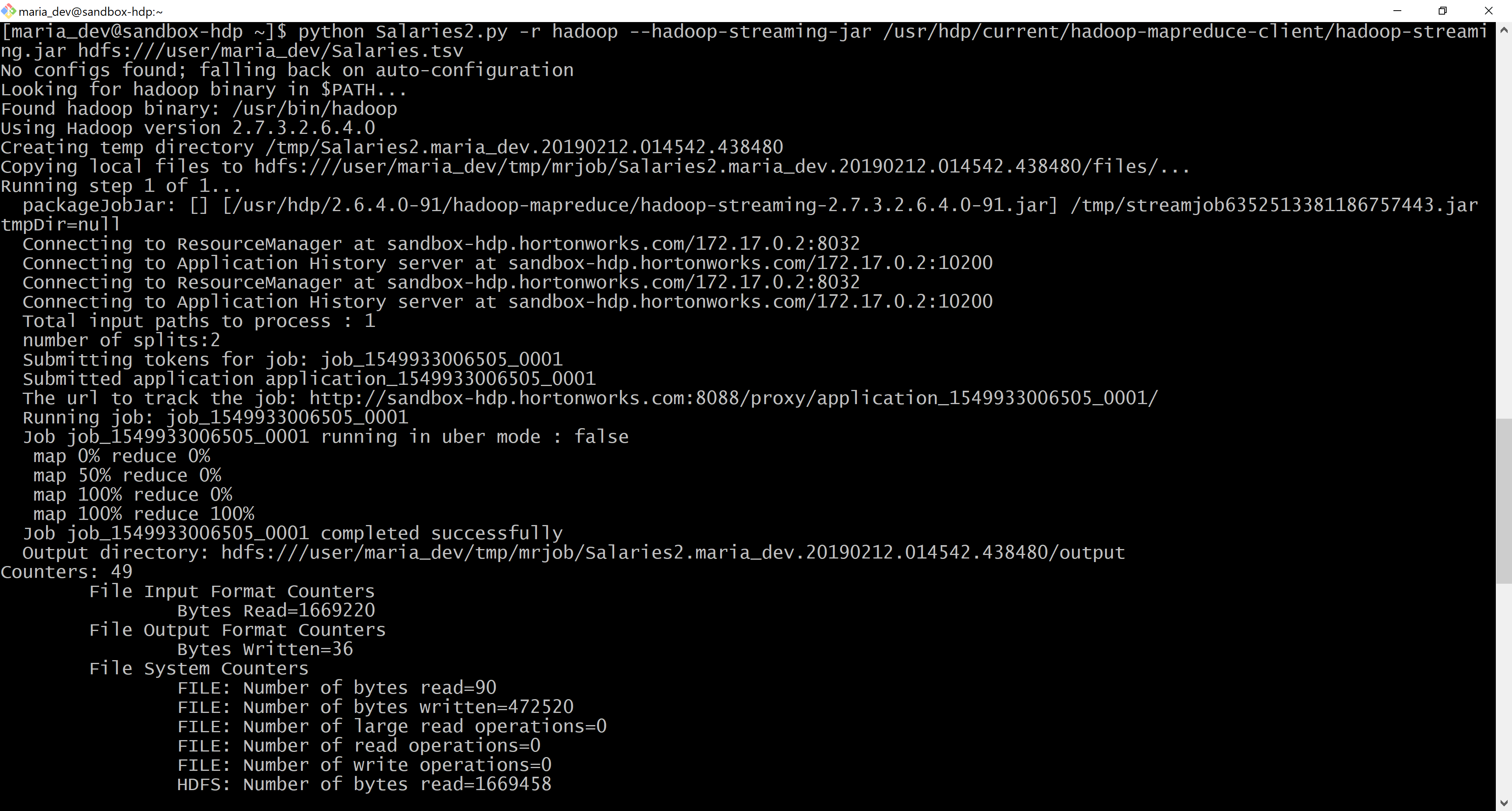
def reducer(self, jobTitle, counts):

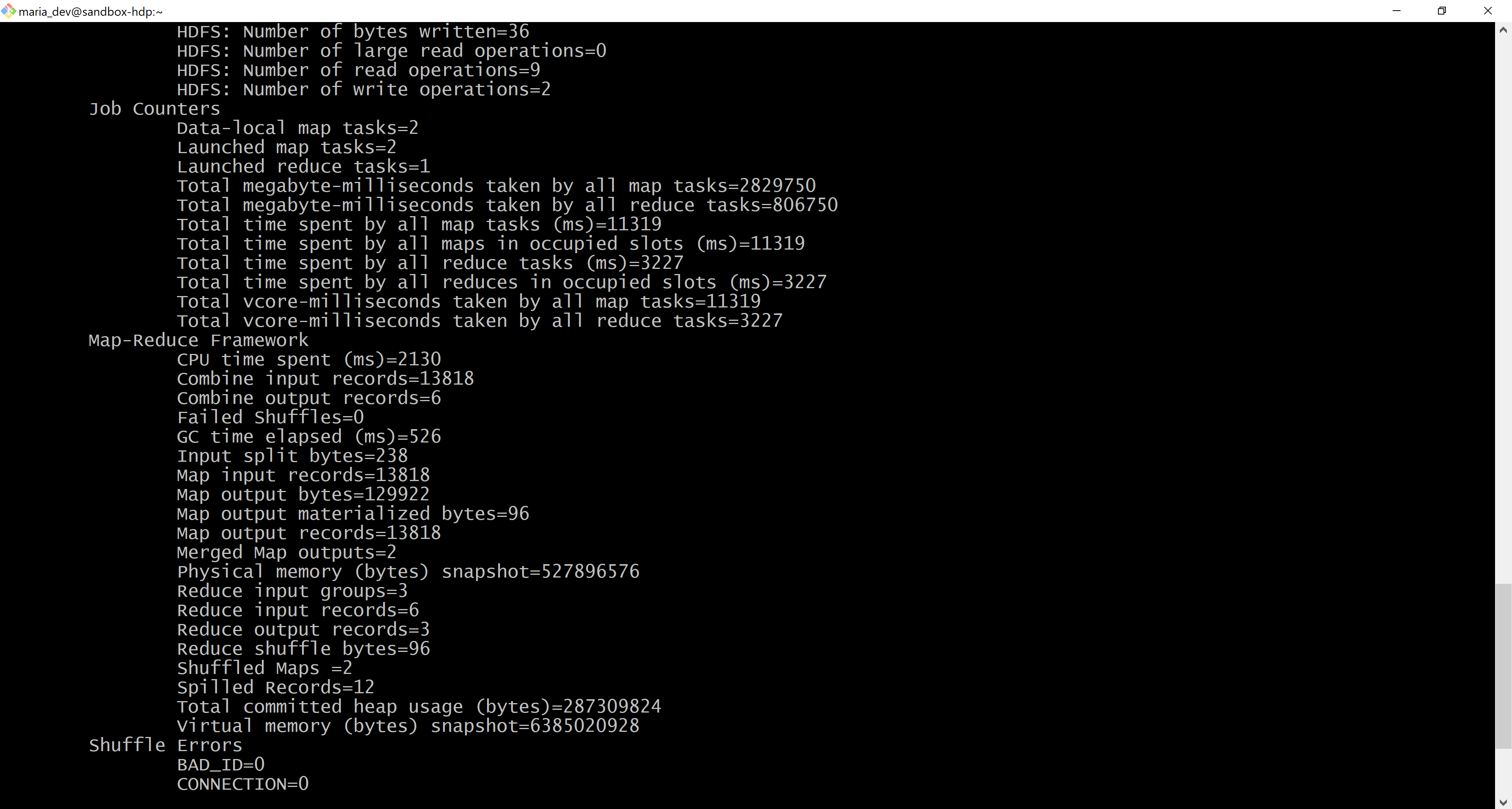
yield jobTitle, sum(counts)

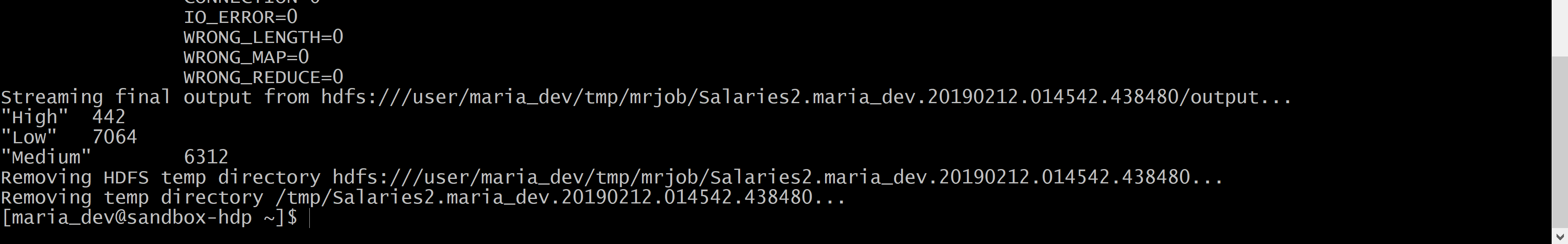
if \_\_name\_\_ == &#39;\_\_main\_\_&#39;:

MRSalaries.run()

**Output:**







**Question 12:**

Review the slides 17-22 in lecture notes Module 3b. Now write a program to perform the task of outputting a count of the number of movies each user (identified via their user id) reviewed.

Output might look something like the following:

186: 2

192: 2

112: 1

etc.

Submit a copy of this program and a screen shot of the results of the program’s execution (only 10 lines or so of the result) as the output of your assignment.

**Program:**

**Ratings.py**

from mrjob.job import MRJob

class MRRatings(MRJob):

def mapper(self, \_, line):

(userid, movieid, rating, timestamp) = line.split(',')

yield userid, 1

def combiner(self, userid, counts):

yield userid, sum(counts)

def reducer(self, userid, counts):

yield userid, sum(counts)

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

MRRatings.run()

**Output:**

