Name: Xinrun Zhang

Date: 15/10/2018

Task1

1. Test mapper.py and reducer.py:

```
zhangxinrun@ubuntu:~/handson-2/task1$ cat pg20417.txt | ./mapper.py |sort| ./red
ucer.py
('the', 9028)
('of', 5530)
('and', 2886)
('a', 2715)
('in', 2336)
('to', 2247)
('is', 2120)
('it', 1383)
('that', 1349)
('are', 947)
```

2. Make a new directory in Hadoop cluster:

```
zhangxinrun@ubuntu:~$ hdfs dfs -mkdir /user/bigdata/handson-2/task1
```

3. Copy the pg20417.txt file into Hadoop cluster:

```
zhangxinrun@ubuntu:~$ hdfs dfs -copyFromLocal /home/zhangxinrun/handson-2/task1/
pg20417.txt /user/bigdata/handson-2/task1
```

4. Run the Hadoop job and save the output into a new folder:

```
zhangxinrun@ubuntu:~$ hadoop jar /usr/local/hadoop/hadoop-streaming-2.9.1.jar -m
apper /home/zhangxinrun/handson-2/task1/mapper.py -reducer /home/zhangxinrun/han
dson-2/task1/reducer.py -input /user/bigdata/handson-2/task1/pg20417.txt -output
 /user/bigdata/handson-2/task1/output
```

5. Result:

   The Hadoop job is running

```
18/10/15 20:35:07 INFO mapreduce.JobSubmitter: Submitting tokens for job: jo
b_1539659905077_0001
18/10/15 20:35:08 INFO impl.YarnClientImpl: Submitted application applicatio
n_1539659905077_0001
18/10/15 20:35:08 INFO mapreduce.Job: The url to track the job: http://ubunt
u:8088/proxy/application_1539659905077_0001/
18/10/15 20:35:08 INFO mapreduce.Job: Running job: job_1539659905077_0001
18/10/15 20:35:25 INFO mapreduce.Job: Job job_1539659905077_0001 running in
uber mode : false
18/10/15 20:35:25 INFO mapreduce.Job:  map 0% reduce 0%
18/10/15 20:35:47 INFO mapreduce.Job:  map 100% reduce 0%
18/10/15 20:35:59 INFO mapreduce.Job:  map 100% reduce 100%
18/10/15 20:36:02 INFO mapreduce.Job: Job job_1539659905077_0001 completed s
uccessfully
```

   Take a look at Hadoop output fold and the outputs are showed:

```
zhangxinrun@ubuntu:~$ hdfs dfs -ls /user/bigdata/handson-2/task1/output
```

```
Found 2 items
-rw-r--r--   1 zhangxinrun supergroup          0 2018-10-15 20:35 /user/bigd
ata/handson-2/task1/output/_SUCCESS
-rw-r--r--   1 zhangxinrun supergroup        143 2018-10-15 20:35 /user/bigd
ata/handson-2/task1/output/part-00000
```

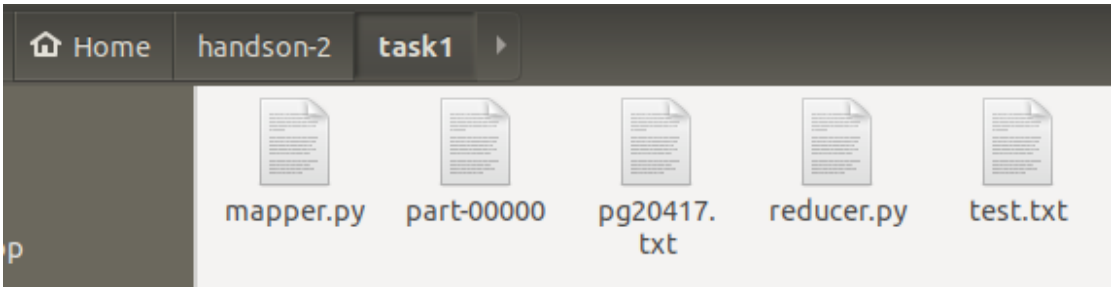
Use –cat command to execute output. The outputs are exactly same as test results:

```
zhangxinrun@ubuntu:~$ hdfs dfs -cat /user/bigdata/handson-2/task1/output/par
t-00000
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authenticat
ion.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop
-auth-2.9.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.had
oop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal ref
lective access operations
WARNING: All illegal access operations will be denied in a future release
('the', 9028)
('of', 5530)
('and', 2886)
('a', 2715)
('in', 2336)
('to', 2247)
('is', 2120)
('it', 1383)
('that', 1349)
('are', 947)
```

Download outputs into local file system.

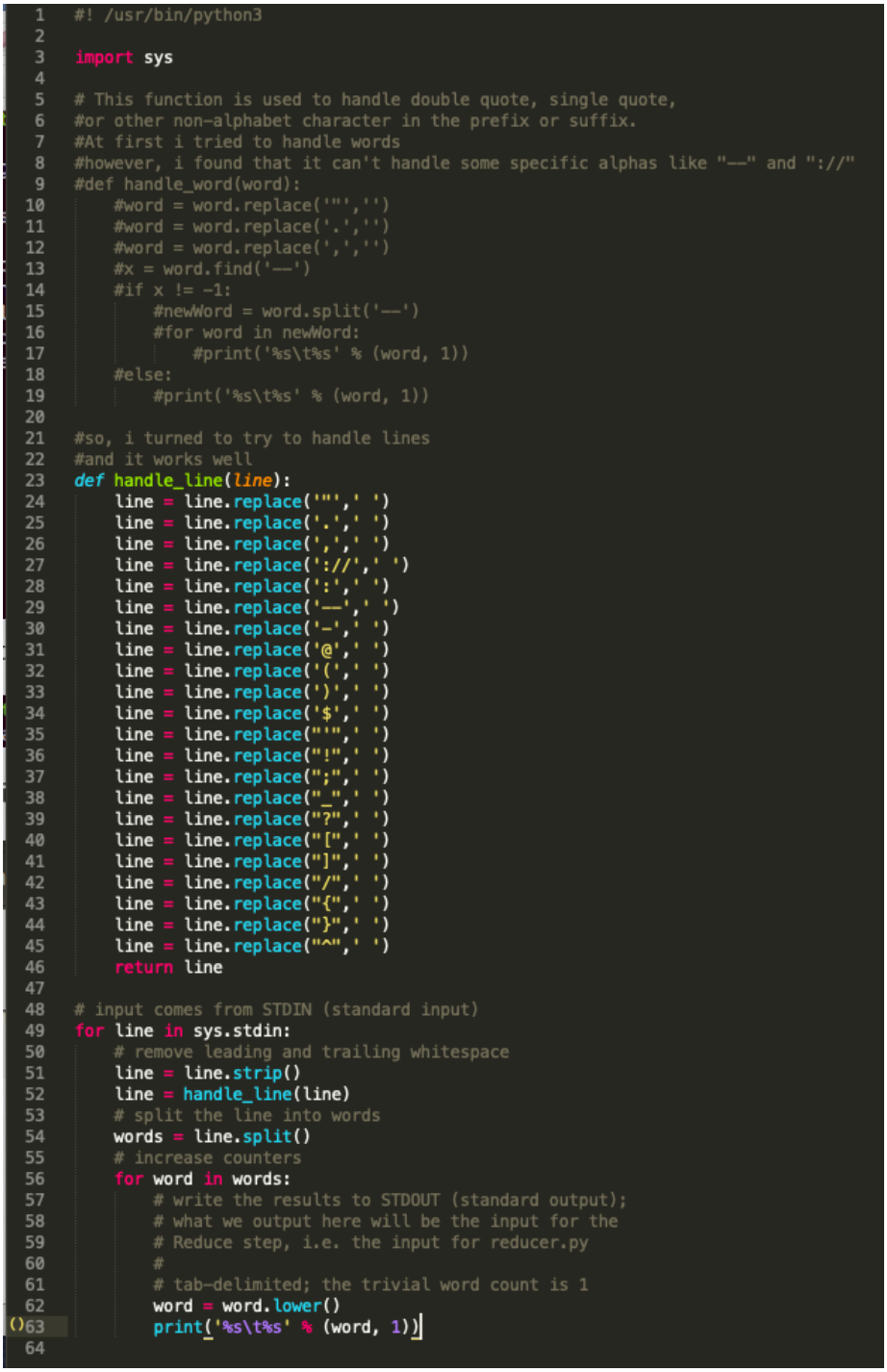
```
zhangxinrun@ubuntu:~$ hdfs dfs -get /user/bigdata/handson-2/task1/output/par
t-00000 /home/zhangxinrun/handson-2/task1
```

the file in local file system, it's a .txt file:



6. Code:

Mapper.py:



```
#! /usr/bin/python3

import sys

# This function is used to handle double quote, single quote,
#or other non-alphabet character in the prefix or suffix.
#At first i tried to handle words
#however, i found that it can't handle some specific alphas like "--" and "://"
#def handle_word(word):
    #word = word.replace('"','')
    #word = word.replace('.','')
    #word = word.replace(',','')
    #x = word.find('--')
    #if x != -1:
        #newWord = word.split('--')
        #for word in newWord:
            #print('%s\t%s' % (word, 1))
    #else:
        #print('%s\t%s' % (word, 1))

#so, i turned to try to handle lines
#and it works well
def handle_line(line):
    line = line.replace('"',' ')
    line = line.replace('.',' ')
    line = line.replace(',',' ')
    line = line.replace('://',' ')
    line = line.replace(':',' ')
    line = line.replace('--',' ')
    line = line.replace('-',' ')
    line = line.replace('@',' ')
    line = line.replace('(',' ')
    line = line.replace(')',' ')
    line = line.replace('$',' ')
    line = line.replace("'",' ')
    line = line.replace("!",' ')
    line = line.replace(";",' ')
    line = line.replace("_",' ')
    line = line.replace("?",' ')
    line = line.replace("[",' ')
    line = line.replace("]",' ')
    line = line.replace("/",' ')
    line = line.replace("{",' ')
    line = line.replace("}",' ')
    line = line.replace("^",' ')
    return line

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    line = handle_line(line)
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        word = word.lower()
        print('%s\t%s' % (word, 1))

```

I defined a handle_line() function to handle the alphas in each line. It replaced alphas with a blank. Then the words in each line were separated naturally and clearly.

Reducer.py:

```python
#!/usr/bin/python3
"""reducer.py"""

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None
newDict = {}

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to newDict{}
            newDict[current_word] = current_count
        current_count = count
        current_word = word

# put the last word into newDict{}
if current_word == word:
    newDict[current_word] = current_count

#sort the newDict{} and put the result into sorted_Dict{}
#output top 10 values and keys
sorted_Dict = sorted(newDict.items(),key = lambda item:item[1],reverse = 1)
m = 0
while m < 10:
    print(sorted_Dict[m])
    m += 1
```

To display top 10 frequency words, I saved the outputs of reducer in a dictionary newDict{} and sorted it. After that, I displayed the first ten keys and values of dictionary.

Task2

1. Test mapper.py and reducer.py

```
zhangxinrun@ubuntu:~/handson-2/task2$ cat purchases.txt | ./mapper.py |sort| ./reducer.py
Albuquerque     149.8976161356743
Anaheim 146.42173162239143
Anchorage       140.5401075683234
Arlington       141.86862165937328
Atlanta 135.36741044462065
Aurora  147.24716460673767
Austin  136.57540721909703
Bakersfield     164.6138453204633
Baltimore       159.90480134406775
Baton Rouge     124.41992734361764
Birmingham      159.64722908498777
Boise   137.5190310123468
Boston  152.80345090576012
```

2. Make a new directory in Hadoop cluster:

```
zhangxinrun@ubuntu:~/handson-2/task2$ cd
zhangxinrun@ubuntu:~$ hdfs dfs -mkdir /user/bigdata/handson-2/task2
```

3. Copy the purchases.txt into Hadoop cluster:

```
zhangxinrun@ubuntu:~$ hdfs dfs -copyFromLocal /home/zhangxinrun/handson-2/task2/purchases.t
xt /user/bigdata/handson-2/task2
```

4. Run the Hadoop job and save the output into a new folder:

```
zhangxinrun@ubuntu:~$ hadoop jar /usr/local/hadoop/hadoop-streaming-2.9.1.jar -mapper /home
/zhangxinrun/handson-2/task2/mapper.py -reducer /home/zhangxinrun/handson-2/task2/reducer.p
y -input /user/bigdata/handson-2/task2/purchases.txt -output /user/bigdata/handson-2/task2/
output
```

5. Result:

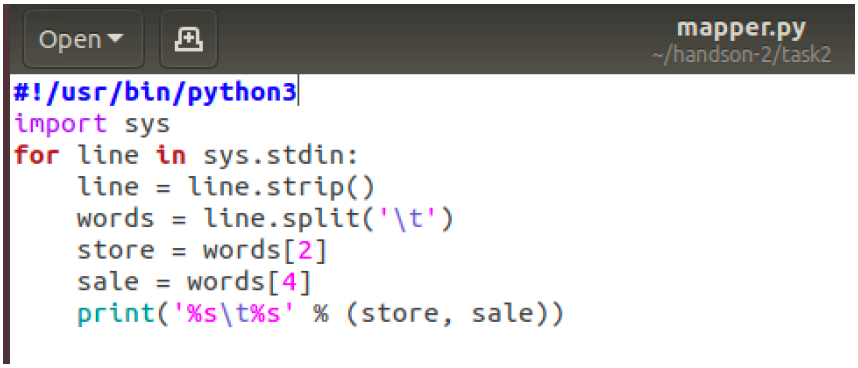
```
18/10/29 13:21:56 INFO mapreduce.Job: Job job_1540842225502_0002 running in uber mode : fal
se
18/10/29 13:21:56 INFO mapreduce.Job:  map 0% reduce 0%
18/10/29 13:22:11 INFO mapreduce.Job:  map 100% reduce 0%
18/10/29 13:22:20 INFO mapreduce.Job:  map 100% reduce 100%
18/10/29 13:22:21 INFO mapreduce.Job: Job job_1540842225502_0002 completed successfully
18/10/29 13:22:22 INFO mapreduce.Job: Counters: 49
```

```
zhangxinrun@ubuntu:~$ hdfs dfs -ls /user/bigdata/handson-2/task2/output
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.Kerber
osUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.1.jar) to method sun
.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.au
thentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access
operations
WARNING: All illegal access operations will be denied in a future release
Found 2 items
-rw-r--r--   1 zhangxinrun supergroup          0 2018-10-29 13:22 /user/bigdata/handson-2/t
ask2/output/_SUCCESS
-rw-r--r--   1 zhangxinrun supergroup       2911 2018-10-29 13:22 /user/bigdata/handson-2/t
ask2/output/part-00000
```

```
zhangxinrun@ubuntu:~$ hdfs dfs -cat /user/bigdata/handson-2/task2/output/part-00000
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.Kerber
osUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.9.1.jar) to method sun
.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.au
thentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access
operations
WARNING: All illegal access operations will be denied in a future release
Albuquerque     142.63462259996976
Anaheim 144.69636714877205
Anchorage       139.4710981021323
Arlington       141.46898374025895
Atlanta 134.70971367024174
Aurora  141.16462827405482
Austin  130.57552668958868
Bakersfield     156.2696982218905
```

6. Code:

mapper.py:



reducer.py:

```
#!/usr/bin/python3
import sys
import math
delta1 = int(0)
delta2 = int(0)
M2 = int(0)
variance = int(0)
(current_store, tot_sales, count) = (None, 0.0, int(0))
for line in sys.stdin:
    (store, sale) = line.split('\t')
    if current_store and store != current_store:
        print('%s\t%s' % (current_store, SD))
        delta1 = int(0)
        delta2 = int(0)
        M2 = int(0)
        variance = int(0)
        (current_store, tot_sales) = (store, float(sale))
        count = 1
    else:
        mean = tot_sales / (count + 1)
        delta1 = float(sale) - mean
        mean = mean + delta1 / (count + 1)
        delta2 = float(sale) - mean
        M2 = M2 + delta1 * delta2
        variance = M2 / (count + 1)
        SD = variance ** 0.5
        (current_store, tot_sales, count) = (store, tot_sales + float(sale), count + 1)
      # print('%s\t%s' % (current_store, SD))
     #   print('%s\t%s\t%s\t%s' % (current_store, tot_sales, SD, count))
if current_store:
    print('%s\t%s' % (current_store, SD))
```

Task3

1. Test mapper.py and reducer.py

```
zhangxinrun@ubuntu:~/handson-2/task3$ cat test.txt | ./mapper.py | sort | ./red
ucer.py
148.224.17.21   1
160.69.35.16    2
198.31.73.63    5
204.255.215.49  7
gandalf.slip.cc.uq.oz.au        3
ix-al12-19.ix.netcom.com        1
ngriffin.itc.gu.edu.au  11
ns.okc.com      44
sydney-ts-38.nstn.ca    1
xslip48.csrv.uidaho.edu 12
```

2. Make a new directory in Hadoop cluster

```
zhangxinrun@ubuntu:~/handson-2/task3$ hdfs dfs -mkdir /user/bigdata/handson-2/task3
```

3. Copy the epa-http.txt into the directory:

```
zhangxinrun@ubuntu:~/handson-2/task3$ cd
zhangxinrun@ubuntu:~$ hdfs dfs -copyFromLocal /home/zhangxinrun/handson-2/task3/epa-http.
txt /user/bigdata/handson-2/task3
```

4. Run the Hadoop job and save the output into a new folder:

```
zhangxinrun@ubuntu:~$ hadoop jar /usr/local/hadoop/hadoop-streaming-2.9.1.jar -mapper /ho
me/zhangxinrun/handson-2/task3/mapper.py -reducer /home/zhangxinrun/handson-2/task3/reduc
er.py -input /user/bigdata/handson-2/task3/epa-http.txt -output /user/bigdata/handson-2/t
ask3/output
```

5. Result:

```
18/10/29 12:59:02 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/
application_1540842225502_0001/
18/10/29 12:59:02 INFO mapreduce.Job: Running job: job_1540842225502_0001
18/10/29 12:59:19 INFO mapreduce.Job: Job job_1540842225502_0001 running in uber mode : f
alse
18/10/29 12:59:19 INFO mapreduce.Job:  map 0% reduce 0%
18/10/29 12:59:39 INFO mapreduce.Job:  map 100% reduce 0%
18/10/29 12:59:49 INFO mapreduce.Job:  map 100% reduce 100%
18/10/29 12:59:51 INFO mapreduce.Job: Job job_1540842225502_0001 completed successfully
18/10/29 12:59:51 INFO mapreduce.Job: Counters: 49
```

```
Found 2 items
-rw-r--r--   1 zhangxinrun supergroup          0 2018-10-29 12:59 /user/bigdata/handson-2
/task3/output/_SUCCESS
-rw-r--r--   1 zhangxinrun supergroup      50792 2018-10-29 12:59 /user/bigdata/handson-2
/task3/output/part-00000
```

```
www.api.org     1
www.cfe.cornell.edu     6
www.hud.gov     2
wwwproxy.ac.il  6
wwwproxy.sanders.com    19
wwwproxy2.ca.sandia.gov 31
wyndmoor1-17.slip.netaxs.com    14
wzb145.wz-berlin.de     2
x121-8.dehs.umn.edu     16
x229-22.ppath.umn.edu   37
xs1.xs4all.nl   24
xslip48.csrv.uidaho.edu 5
xyplex4-1-20.ucs.indiana.edu    22
y.nsf.gov       1
yaqut.ar.utexas.edu     9
yhqfm.yokogawa.co.jp    1
yukonho.cs.caltech.edu  8
yyj-ppp-13.cyberstore.ca        14
yyj-ppp-14.cyberstore.ca        4
yyj-ppp-8.cyberstore.ca 1
zeus.esy.com    86
zorba.wictok1.epa.gov   3
zuni.r09.epa.gov        20
```

6. Code:

mapper.py:

Open mapper.py ~/handson-2/task3

```
#! /usr/bin/python3
import sys

x = 0
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    attributes = line.split()
    #x = attributes[3].find('.gif')
    #y = attributes[3].find('.htm')
    #z = attributes[3].find('.xbm')
    #if x!= -1 or y!= -1 or z!= -1:
        #print('%s\t%s' % (attributes[0],attributes[3]))
    print('%s\t%s' % (attributes[0],attributes[3]))
```

reducer.py:

```
#!/usr/bin/python3
"""reducer.py"""

from operator import itemgetter
import sys

current_hostname = None
current_filename = None
hostname = None
filename = None
count = 1

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    hostname, filename = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # by key (here: word) before it is passed to the reducer
    if current_hostname == hostname and current_filename != filename:
        count += 1
    else:
        if current_hostname and current_hostname != hostname:
            # write result to ouput
            print('%s\t%s' % (current_hostname, count))
        count = 1
        current_hostname = hostname
        current_filename = filename

# put the last word into newDict{}
if current_hostname == hostname:
    print('%s\t%s' % (current_hostname, count))
```