Assignment 01: Determine the wordcount

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

Determine the word count of the given Amazon dataset:

DESCRIPTION

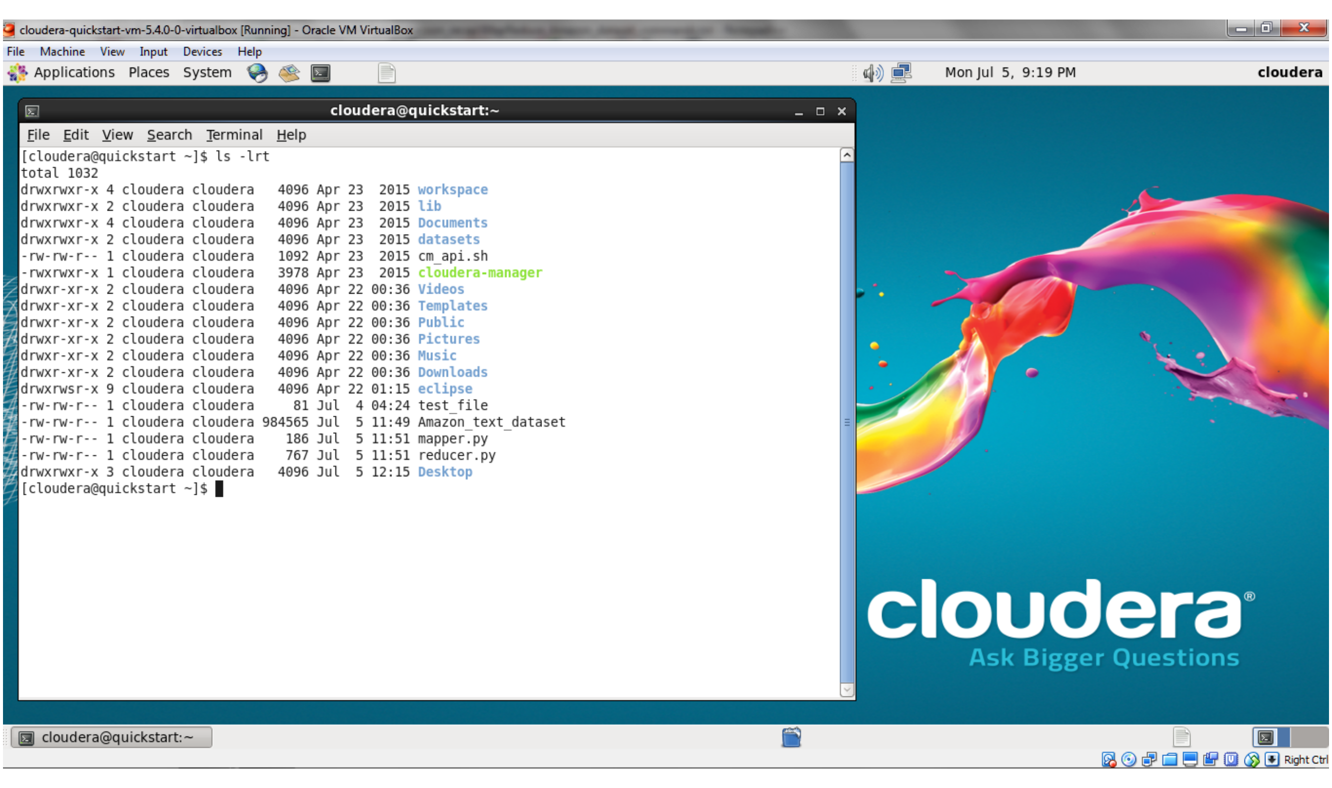
Problem:

- Create a MapReduce program to determine the word count of the Amazon dataset.
- Submit the MapReduce task to HDFS and run it.
- Verify the output.

Create a MapReduce program to determine the word count of the Amazon dataset

Copy the Dataset to Home Directory

- Copy the Amzon_text_dataset to cloudera Home Directory.
- Open the Terminal Application.
- Enter "ls -lrt" to list the folders and files in Home directory.



View the Dataset

- View the dataset using "cat Amazon_text_data" command.

```
[cloudera@quickstart ~]$ cat Amazon_text_dataset
I try not to adjust the volume setting to avoid that I turn off the call button which is situated just below the volume adjustment knob.
So there is no way for me to plug it in here in the US unless I go by a converter.  0
Good case, Excellent value. 1
I thought Motorola made reliable products!.
Battery for Motorola Razr.
Great for the jawbone.  1
When I got this item it was larger than I thought It was the size of a small video game but bulky.
(I looked for one that specifically said DCU-60 and said it supported USB 1 and 2.
The first time it was turned on the screen display began to flicker.
In some programs clicking it is the same as hitting the return key.
Tied to charger for conversations lasting more than 45 minutes.MAJOR PROBLEMS!! 0
The mic is great.   1
What happened was that I only had like 70 contacts but the SIM memory said I had 250, and that all my SIM memory was full.
I have to jiggle the plug to get it to line up right to get decent volume.  0
I bought five of thes for less than five dollars.
If you have several dozen or several hundred contacts, then imagine the fun of sending each of them one by one. 0
I had the two year contract for this service.
```

Create the Mapper code

- Create mapper python program in Vi editor using "Vi mapper.py" command.
- Use "cat mapper.py" to view the contents of the file.

```
[cloudera@quickstart ~]$ cat mapper.py
#!/usr/bin/env python

import sys

for line in sys.stdin:
    #remove loading and trailing spaces
    line = line.strip()
    words = line.split()

    for word in words:
        print("%s\t%s"%(word,1))
```

Create the Reducer code

- Create reducer python program in Vi editor using "Vi reducer.py" command.
- Use "cat reducer.py" to view the contents of the file.

```
[cloudera@quickstart ~]$ cat reducer.py
import sys
from operator import itemgetter

current_word = None
current_count = 0
word = None

#input comes from console
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:
        #ignore if the count is not a number
        continue

    #use the sorted map output
    if current_word == word:
        current_count+=count
    else:
        if current_word:
            #write result in console
            print("%s\t%s" %(current_word, current_count))
        current_count = count
        current_word = word

#finally output the word to the console
if current_word == word:
    print("%s\t%s" %(current_word,current_count))
```

Execute the Mapper program over the Amazon_text_dataset

- Use "cat Amazon_text_dataset | python mapper.py" commant to execute the Mapper program over the Amazon_text_dataset.

```
[cloudera@quickstart ~]$ cat Amazon_text_dataset | python mapper.py
I   1
try 1
not 1
to  1
adjust  1
the 1
volume  1
setting 1
to  1
avoid   1
that    1
I   1
turn    1
off 1
the 1
call    1
button  1
which   1
is  1
situated    1
```

Submit the MapReduce task to HDFS and run it.

Copy the Amazon_text_dataset file to HDFS location

- Use "hdfs dfs -put Amazon_text_dataset /user/cloudera" to copy the "Amazon_text_dataset" to HDFS location.
- Use "hdfs dfs -ls /user/cloudera" to list the folders and files in HDFS location.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera
Found 7 items
-rw-r--r--   1 cloudera cloudera     984565 2021-07-05 11:55 /user/cloudera/Amazon_text_dataset
-rw-r--r--   1 cloudera cloudera         81 2021-07-04 03:34 /user/cloudera/test_file
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 03:46 /user/cloudera/wc_output01
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 04:20 /user/cloudera/wc_output02
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 04:31 /user/cloudera/wc_output03
drwxr-xr-x   - cloudera cloudera          0 2021-07-05 12:11 /user/cloudera/wc_output04
drwxr-xr-x   - cloudera cloudera          0 2021-07-05 21:14 /user/cloudera/wc_output05
```

Submit the MapReduce Task

- Using below command, sumbit the MapReduce task, add the below line in bash.
- Once the task is submitted, it runs the mapper first and then rins the reducer program over the dataset.

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/cloudera/mapper.py /home/cloudera/reducer.py -mapper "python mapper.py" -reducer "python reducer.py" -input /user/cloudera/Amazon_text_dataset -output /user/cloudera/wc_output06

```
[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/cloudera/mapper.py /home/cloudera/reducer.py
        -mapper "python mapper.py" -reducer "python reducer.py" -input /user/cloudera/Amazon_text_dataset -output /user/cloudera/wc_output06
21/07/05 21:27:06 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/home/cloudera/mapper.py, /home/cloudera/reducer.py] [/usr/lib/hadoop-mapreduce/hadoop-streaming-2.6.0-cdh5.4.0.jar] /tmp/streamjob181468697571
21/07/05 21:27:15 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/07/05 21:27:19 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/07/05 21:27:23 INFO mapred.FileInputFormat: Total input paths to process : 1
21/07/05 21:27:23 INFO mapreduce.JobSubmitter: number of splits:2
21/07/05 21:27:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1625542583719_0002
21/07/05 21:27:27 INFO impl.YarnClientImpl: Submitted application application_1625542583719_0002
21/07/05 21:27:27 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1625542583719_0002/
21/07/05 21:27:27 INFO mapreduce.Job: Running job: job_1625542583719_0002
21/07/05 21:28:07 INFO mapreduce.Job: Job job_1625542583719_0002 running in uber mode : false
21/07/05 21:28:07 INFO mapreduce.Job:  map 0% reduce 0%
21/07/05 21:28:49 INFO mapreduce.Job:  map 18% reduce 0%
21/07/05 21:28:51 INFO mapreduce.Job:  map 44% reduce 0%
21/07/05 21:28:53 INFO mapreduce.Job:  map 60% reduce 0%
21/07/05 21:28:54 INFO mapreduce.Job:  map 67% reduce 0%
21/07/05 21:28:58 INFO mapreduce.Job:  map 100% reduce 0%
21/07/05 21:29:27 INFO mapreduce.Job:  map 100% reduce 77%
21/07/05 21:29:30 INFO mapreduce.Job:  map 100% reduce 100%
21/07/05 21:29:31 INFO mapreduce.Job: Job job_1625542583719_0002 completed successfully
21/07/05 21:29:32 INFO mapreduce.Job: Counters: 49
    File System Counters
        FILE: Number of bytes read=1693573
        FILE: Number of bytes written=3728548
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
```

```
        Combine input records=0
        Combine output records=0
        Reduce input groups=18818
        Reduce shuffle bytes=1693579
        Reduce input records=180753
        Reduce output records=18818
        Spilled Records=361506
        Shuffled Maps =2
        Failed Shuffles=0
        Merged Map outputs=2
        GC time elapsed (ms)=923
        CPU time spent (ms)=9430
        Physical memory (bytes) snapshot=550928384
        Virtual memory (bytes) snapshot=4511481856
        Total committed heap usage (bytes)=312680448
    Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
    File Input Format Counters
        Bytes Read=987899
    File Output Format Counters
        Bytes Written=191772
21/07/05 21:29:32 INFO streaming.StreamJob: Output directory: /user/cloudera/wc_output06
```

Verify the output

- Verify the output folder present in HDFS loacation using below command.
  - hdfs dfs -ls /user/clouder

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera
Found 8 items
-rw-r--r--   1 cloudera cloudera     984565 2021-07-05 11:55 /user/cloudera/Amazon_text_dataset
-rw-r--r--   1 cloudera cloudera         81 2021-07-04 03:34 /user/cloudera/test_file
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 03:46 /user/cloudera/wc_output01
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 04:20 /user/cloudera/wc_output02
drwxr-xr-x   - cloudera cloudera          0 2021-07-04 04:31 /user/cloudera/wc_output03
drwxr-xr-x   - cloudera cloudera          0 2021-07-05 12:11 /user/cloudera/wc_output04
drwxr-xr-x   - cloudera cloudera          0 2021-07-05 21:14 /user/cloudera/wc_output05
drwxr-xr-x   - cloudera cloudera          0 2021-07-05 21:29 /user/cloudera/wc_output06
```

- To check the Output on HDFS use the below comment.
  - hdfs dfs -ls /user/cloudera/wc_output06

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/wc_output06
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2021-07-05 21:29 /user/cloudera/wc_output06/_SUCCESS
-rw-r--r--   1 cloudera cloudera     191772 2021-07-05 21:29 /user/cloudera/wc_output06/part-00000
```

- To View the part file use below command
  - hdfs dfs -cat /user/cloudera/wc_output06/part*

```
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/wc_output06/part*
!   17
!!  4
!!! 2
!!!!!   1
!!!!!.  1
!!!.    1
!!. 1
!$$$    1
!,  1
!.  5
!...I   1
!...ridiculous, 1
!1) 1
!1. 1
!2. 1
!At 1
!BURN   1
!Do 1
!GONNA  1
!Gonna  1
!Hope   1
!I  2
!I've   1
!If 1
```

In [ ]: