Ryan Ballenger
CSCIE63 Big Data Analytics
Assignment 3

Assignment 3 Solution

Table of Contents

Problem 1. Steps
Problem 2. Steps
Problem 3. Steps
Problem 4. Steps
Problem 5. Steps

Problem 1. Results
Problem 2. Results
Problem 3 Results
Problem 4 Results
Problem 5 Results

Problem 1) Modify attached class WordCount.java so that its result excludes the following stop words:

I
a
about
an
are
as
at
be
by
com
for
from
how
in
is
it
of
on
or
that
the
this
to
was
what
when
where
who
will
with
the
www

as well as special characters (dashes, parentases, etc). Stop word lists could be much lomger than this. You do not have to be extremely thoroughful. You would like to get a more or less clean list of ordinary words with the numbers of their occurances. Do not fret. Be reasonable. Perform analysis on te text of James Joyce's Ulysis.

Problem 1. Steps

# Program overview: the stop words, upper or lower case, and any
# words made up of special characters without any alphanumeric characters

# are removed.

\#Note: I renamed my class to **WordCountTwo.java** to prevent running the wrong
# program. The first 20 results are listed in the appendix of this document.

# First I exported the Hadoop classpath to a variable HCP to make the
# compiling commands more convenient

```
[joe@localhost examples]$ export
HCP=/etc/hadoop/conf:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hadoop-hdfs/./:/u
sr/lib/hadoop-hdfs/lib/*:/usr/lib/hadoop-hdfs/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoo
p-yarn/.//*:/usr/lib/hadoop-mapreduce/lib/*:/usr/lib/hadoop-mapreduce/.//*
```

# I compiled my program **WordCountTwo.java**

```
[joe@localhost examples]$ javac -classpath $HCP -d . WordCountTwo.java
```

# I created a jar with my class files for WordCountTwo

```
[joe@localhost examples]$ jar -cvf wordcount.jar org/*
added manifest
adding: org/apache/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/WordCountTwo$TokenizerMapper.class(in =
2636) (out= 1302)(deflated 50%)
adding: org/apache/hadoop/examples/WordCountTwo$IntSumReducer.class(in = 1802)
(out= 754)(deflated 58%)
adding: org/apache/hadoop/examples/WordCountTwo.class(in = 2005) (out=
1058)(deflated 47%)
```

# Finally I successfully ran WordCountTwo on the Ulysses text

```
[joe@localhost examples]$ hadoop jar wordcount.jar
org.apache.hadoop.examples.WordCountTwo ulysses output
16/02/18 18:38:21 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/02/18 18:38:23 INFO input.FileInputFormat: Total input paths to process : 1
16/02/18 18:38:23 INFO mapreduce.JobSubmitter: number of splits:1
16/02/18 18:38:23 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455825540424_0009
16/02/18 18:38:24 INFO impl.YarnClientImpl: Submitted application
application_1455825540424_0009
16/02/18 18:38:24 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455825540424_0009/
```

```
16/02/18 18:38:24 INFO mapreduce.Job: Running job: job_1455825540424_0009
16/02/18 18:38:37 INFO mapreduce.Job: Job job_1455825540424_0009 running in uber
mode : false
16/02/18 18:38:37 INFO mapreduce.Job:  map 0% reduce 0%
16/02/18 18:38:51 INFO mapreduce.Job:  map 100% reduce 0%
16/02/18 18:39:02 INFO mapreduce.Job:  map 100% reduce 100%
16/02/18 18:39:03 INFO mapreduce.Job: Job job_1455825540424_0009 completed
successfully
16/02/18 18:39:03 INFO mapreduce.Job: Counters: 49
	File System Counters
		FILE: Number of bytes read=724071
		FILE: Number of bytes written=1671633
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=1573191
		HDFS: Number of bytes written=526820
		HDFS: Number of read operations=6
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=12330
		Total time spent by all reduces in occupied slots (ms)=7636
		Total time spent by all map tasks (ms)=12330
		Total time spent by all reduce tasks (ms)=7636
		Total vcore-seconds taken by all map tasks=12330
		Total vcore-seconds taken by all reduce tasks=7636
		Total megabyte-seconds taken by all map tasks=12625920
		Total megabyte-seconds taken by all reduce tasks=7819264
	Map-Reduce Framework
		Map input records=33056
		Map output records=201025
		Map output bytes=2103796
		Map output materialized bytes=724071
		Input split bytes=112
		Combine input records=201025
		Combine output records=49998
		Reduce input groups=49998
		Reduce shuffle bytes=724071
		Reduce input records=49998
		Reduce output records=49998
		Spilled Records=99996
```

```
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=428
                CPU time spent (ms)=7730
                Physical memory (bytes) snapshot=322125824
                Virtual memory (bytes) snapshot=5440196608
                Total committed heap usage (bytes)=165810176
        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=1573079
        File Output Format Counters
                Bytes Written=526820
```

Problem 2) Write another MapReduce program which would read the "word count" output of the previous job and order the results by the declaining number of occurances.

Problem 2. Steps

```
# Program overview: It sorts words by count into descending order. My program enters
# the count in as the key in the map phase and uses a custom key sorter to get descending
# order. It then writes out the count and the word in the reduce phase.

# As in #1, I compiled my java program named WordCountThree.java for this
# problem and created a jar with the class file named wordcount.jar

[joe@localhost examples]$ javac -classpath $HCP -d . WordCountThree.java
[joe@localhost examples]$ jar -cvf wordcount.jar org/*
added manifest
adding: org/apache/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/WordCountThree$Reverse.class(in = 643) (out=
355)(deflated 44%)
adding: org/apache/hadoop/examples/WordCountThree.class(in = 2223) (out=
1138)(deflated 48%)
adding: org/apache/hadoop/examples/WordCountThree$TokenizerMapper.class(in =
1837) (out= 754)(deflated 58%)
```

```
adding: org/apache/hadoop/examples/WordCountThree$IntSumReducer.class(in = 2040)
(out= 876)(deflated 57%)
```

# I ran the program **WordCountThree** that reorders the words by number of
# occurrences in descending order. The file with WordCountTwo's output is renamed
# to 1044 and the output of this program is 1051. Additionally, the first 20 results
# are listed in the appendix for references.

```
[joe@localhost examples]$ hadoop jar wordcount.jar
org.apache.hadoop.examples.WordCountThree 1044 1051
16/02/19 19:51:14 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/02/19 19:51:16 INFO input.FileInputFormat: Total input paths to process : 1
16/02/19 19:51:16 INFO mapreduce.JobSubmitter: number of splits:1
16/02/19 19:51:17 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455938417648_0002
16/02/19 19:51:17 INFO impl.YarnClientImpl: Submitted application
application_1455938417648_0002
16/02/19 19:51:17 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455938417648_0002/
16/02/19 19:51:17 INFO mapreduce.Job: Running job: job_1455938417648_0002
16/02/19 19:51:32 INFO mapreduce.Job: Job job_1455938417648_0002 running in uber
mode : false
16/02/19 19:51:32 INFO mapreduce.Job:  map 0% reduce 0%
16/02/19 19:51:44 INFO mapreduce.Job:  map 100% reduce 0%
16/02/19 19:51:57 INFO mapreduce.Job:  map 100% reduce 100%
16/02/19 19:51:58 INFO mapreduce.Job: Job job_1455938417648_0002 completed
successfully
16/02/19 19:51:58 INFO mapreduce.Job: Counters: 49
	File System Counters
		FILE: Number of bytes read=724071
		FILE: Number of bytes written=1672701
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=526933
		HDFS: Number of bytes written=526820
		HDFS: Number of read operations=6
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=10662
		Total time spent by all reduces in occupied slots (ms)=9346
		Total time spent by all map tasks (ms)=10662
		Total time spent by all reduce tasks (ms)=9346
		Total vcore-seconds taken by all map tasks=10662
		Total vcore-seconds taken by all reduce tasks=9346
		Total megabyte-seconds taken by all map tasks=10917888
		Total megabyte-seconds taken by all reduce tasks=9570304
	Map-Reduce Framework
		Map input records=49998
		Map output records=49998
		Map output bytes=624069
		Map output materialized bytes=724071
		Input split bytes=113
		Combine input records=49998
		Combine output records=49998
		Reduce input groups=236
		Reduce shuffle bytes=724071
		Reduce input records=49998
		Reduce output records=49998
		Spilled Records=99996
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		GC time elapsed (ms)=690
		CPU time spent (ms)=5470
		Physical memory (bytes) snapshot=321998848
		Virtual memory (bytes) snapshot=5445128192
		Total committed heap usage (bytes)=165810176
	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=526820
	File Output Format Counters
		Bytes Written=526820
```

Problem 3) Create a program that will find out how many words appear only once, how many twice, three times, four times and so on in James Joyce's Ulysis

Problem 3. Steps

# **WordCountFour** overview: It determines how many words occur once, twice, three
# times, and so on. It enters a word's count as the key and an IntWritable value of one
# during mapping. It then finds the sum of IntWritables in the reducing phase for each
# word count.

# I compile my program named **WordCountFour**, put the class file into a jar.

```
[joe@localhost examples]$ javac -classpath $HCP -d . WordCountFour.java
[joe@localhost examples]$ jar -cvf wordcount.jar org/*added manifest
adding: org/apache/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/WordCountFour.class(in = 2099) (out=
1088)(deflated 48%)
adding: org/apache/hadoop/examples/WordCountFour$TokenizerMapper.class(in =
1834) (out= 752)(deflated 58%)
adding: org/apache/hadoop/examples/WordCountFour$IntSumReducer.class(in = 2032)
(out= 846)(deflated 58%)
```

# I run the mapreduce program with WordCountTwo's output as the input in the file
# firstoutput and this program outputs to the folder named thirtyone

```
[joe@localhost examples]$ hadoop jar wordcount.jar
org.apache.hadoop.examples.WordCountFour firstoutput thirtyone
16/02/18 22:31:59 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/02/18 22:32:01 INFO input.FileInputFormat: Total input paths to process : 1
16/02/18 22:32:01 INFO mapreduce.JobSubmitter: number of splits:1
16/02/18 22:32:01 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455825540424_0031
16/02/18 22:32:02 INFO impl.YarnClientImpl: Submitted application
application_1455825540424_0031
16/02/18 22:32:02 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455825540424_0031/
16/02/18 22:32:02 INFO mapreduce.Job: Running job: job_1455825540424_0031
16/02/18 22:32:15 INFO mapreduce.Job: Job job_1455825540424_0031 running in uber
mode : false
16/02/18 22:32:15 INFO mapreduce.Job:  map 0% reduce 0%
16/02/18 22:32:25 INFO mapreduce.Job:  map 100% reduce 0%
16/02/18 22:32:34 INFO mapreduce.Job:  map 100% reduce 100%
16/02/18 22:32:35 INFO mapreduce.Job: Job job_1455825540424_0031 completed
successfully
16/02/18 22:32:36 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=1966
                FILE: Number of bytes written=228115
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=526937
                HDFS: Number of bytes written=1316
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7996
                Total time spent by all reduces in occupied slots (ms)=6703
                Total time spent by all map tasks (ms)=7996
                Total time spent by all reduce tasks (ms)=6703
                Total vcore-seconds taken by all map tasks=7996
                Total vcore-seconds taken by all reduce tasks=6703
                Total megabyte-seconds taken by all map tasks=8187904
                Total megabyte-seconds taken by all reduce tasks=6863872
        Map-Reduce Framework
                Map input records=49998
                Map output records=49998
                Map output bytes=624069
                Map output materialized bytes=1966
                Input split bytes=117
                Combine input records=49998
                Combine output records=236
                Reduce input groups=236
                Reduce shuffle bytes=1966
                Reduce input records=236
                Reduce output records=236
                Spilled Records=472
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=303
                CPU time spent (ms)=3430
                Physical memory (bytes) snapshot=314617856
                Virtual memory (bytes) snapshot=5445595136
                Total committed heap usage (bytes)=165810176
        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=526820
        File Output Format Counters
                Bytes Written=1316
```

Problem 4) Combine operations of two MapReduce programs in Problems 1 and 3 above into a single program with chained MapReduce jobs.

Problem 4. Steps
# WordCountChained overview: combine solutions from #1 and #3 with mapreduce chaining and a temporary storage file. Delete the temporary file after both jobs complete.

# Note: First 20 results are listed in the appendix.
# I compile and jar the class file for the class **WordCountChained**

```
[joe@localhost examples]$ javac -classpath $HCP -d . WordCountChained.java
[joe@localhost examples]$ jar -cvf wordcount.jar org/*
added manifest
adding: org/apache/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/(in = 0) (out= 0)(stored 0%)
adding: org/apache/hadoop/examples/WordCountChained$TokenizerMapper.class(in =
1822) (out= 744)(deflated 59%)
adding: org/apache/hadoop/examples/WordCountChained$TokenizerMapperOne.class(in
= 2654) (out= 1318)(deflated 50%)
adding: org/apache/hadoop/examples/WordCountChained.class(in = 2895) (out=
1291)(deflated 55%)
adding: org/apache/hadoop/examples/WordCountChained$IntSumReducer.class(in =
1870) (out= 771)(deflated 58%)
adding: org/apache/hadoop/examples/WordCountChained$IntSumReducerOne.class(in =
1820) (out= 760)(deflated 58%)
```

# I run the **WordCountChained** program on the ulysses file and the two mapreduce
# processes output below. Also the output file is named 1109

```
[joe@localhost examples]$ hadoop jar wordcount.jar
org.apache.hadoop.examples.WordCountChained ulysses 1109
16/02/19 20:09:33 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
```

```
16/02/19 20:09:34 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your application
with ToolRunner to remedy this.
16/02/19 20:09:35 INFO input.FileInputFormat: Total input paths to process : 1
16/02/19 20:09:35 INFO mapreduce.JobSubmitter: number of splits:1
16/02/19 20:09:35 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455938417648_0004
16/02/19 20:09:36 INFO impl.YarnClientImpl: Submitted application
application_1455938417648_0004
16/02/19 20:09:36 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455938417648_0004/
16/02/19 20:09:36 INFO mapreduce.Job: Running job: job_1455938417648_0004
16/02/19 20:09:50 INFO mapreduce.Job: Job job_1455938417648_0004 running in uber
mode : false
16/02/19 20:09:50 INFO mapreduce.Job:  map 0% reduce 0%
16/02/19 20:10:06 INFO mapreduce.Job:  map 46% reduce 0%
16/02/19 20:10:09 INFO mapreduce.Job:  map 67% reduce 0%
16/02/19 20:10:14 INFO mapreduce.Job:  map 100% reduce 0%
16/02/19 20:10:26 INFO mapreduce.Job:  map 100% reduce 100%
16/02/19 20:10:27 INFO mapreduce.Job: Job job_1455938417648_0004 completed
successfully
16/02/19 20:10:28 INFO mapreduce.Job: Counters: 49
	File System Counters
		FILE: Number of bytes read=724071
		FILE: Number of bytes written=1671399
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=1573191
		HDFS: Number of bytes written=526820
		HDFS: Number of read operations=6
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=22381
		Total time spent by all reduces in occupied slots (ms)=9802
		Total time spent by all map tasks (ms)=22381
		Total time spent by all reduce tasks (ms)=9802
		Total vcore-seconds taken by all map tasks=22381
		Total vcore-seconds taken by all reduce tasks=9802
		Total megabyte-seconds taken by all map tasks=22918144
		Total megabyte-seconds taken by all reduce tasks=10037248
```

```
        Map-Reduce Framework
                Map input records=33056
                Map output records=201025
                Map output bytes=2103796
                Map output materialized bytes=724071
                Input split bytes=112
                Combine input records=201025
                Combine output records=49998
                Reduce input groups=49998
                Reduce shuffle bytes=724071
                Reduce input records=49998
                Reduce output records=49998
                Spilled Records=99996
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=521
                CPU time spent (ms)=12040
                Physical memory (bytes) snapshot=321773568
                Virtual memory (bytes) snapshot=5440188416
                Total committed heap usage (bytes)=165810176
        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=1573079
        File Output Format Counters
                Bytes Written=526820
16/02/19 20:10:28 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/02/19 20:10:28 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your application
with ToolRunner to remedy this.
16/02/19 20:10:28 INFO input.FileInputFormat: Total input paths to process : 1
16/02/19 20:10:28 INFO mapreduce.JobSubmitter: number of splits:1
16/02/19 20:10:29 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455938417648_0005
16/02/19 20:10:29 INFO impl.YarnClientImpl: Submitted application
application_1455938417648_0005
16/02/19 20:10:29 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455938417648_0005/
```

```
16/02/19 20:10:29 INFO mapreduce.Job: Running job: job_1455938417648_0005
16/02/19 20:10:44 INFO mapreduce.Job: Job job_1455938417648_0005 running in uber
mode : false
16/02/19 20:10:44 INFO mapreduce.Job:  map 0% reduce 0%
16/02/19 20:11:03 INFO mapreduce.Job:  map 100% reduce 0%
16/02/19 20:11:15 INFO mapreduce.Job:  map 100% reduce 100%
16/02/19 20:11:16 INFO mapreduce.Job: Job job_1455938417648_0005 completed
successfully
16/02/19 20:11:16 INFO mapreduce.Job: Counters: 49
	File System Counters
		FILE: Number of bytes read=2366
		FILE: Number of bytes written=227967
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=526939
		HDFS: Number of bytes written=1388
		HDFS: Number of read operations=6
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=17373
		Total time spent by all reduces in occupied slots (ms)=9578
		Total time spent by all map tasks (ms)=17373
		Total time spent by all reduce tasks (ms)=9578
		Total vcore-seconds taken by all map tasks=17373
		Total vcore-seconds taken by all reduce tasks=9578
		Total megabyte-seconds taken by all map tasks=17789952
		Total megabyte-seconds taken by all reduce tasks=9807872
	Map-Reduce Framework
		Map input records=49998
		Map output records=49998
		Map output bytes=399984
		Map output materialized bytes=2366
		Input split bytes=119
		Combine input records=49998
		Combine output records=236
		Reduce input groups=236
		Reduce shuffle bytes=2366
		Reduce input records=236
		Reduce output records=236
		Spilled Records=472
```

```
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=303
                CPU time spent (ms)=8920
                Physical memory (bytes) snapshot=316198912
                Virtual memory (bytes) snapshot=5440196608
                Total committed heap usage (bytes)=165810176
        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=526820
        File Output Format Counters
                Bytes Written=1388
```

Problem 5) Move attached Inverter.java class from old MapReduce API to the new API. Demonstrate that new and old clas produce the same result. Use patent data set to demonstrate your work.

Problem 5. Steps

# I compile the **InverterNewAPI.java** into a class and create a jar
# for the class file. The jar is named wordcount.jar which is undescriptive
# but will function to run the program.

```
[joe@localhost inversion]$ javac -classpath $HCP -d . InverterNewAPI.java
[joe@localhost inversion]$ jar -cvf wordcount.jar edu/*
added manifest
adding: edu/hu/(in = 0) (out= 0)(stored 0%)
adding: edu/hu/bigdata/(in = 0) (out= 0)(stored 0%)
adding: edu/hu/bigdata/InverterNewAPI.class(in = 2352) (out= 1106)(deflated 52%)
adding: edu/hu/bigdata/InverterNewAPI$Reduce.class(in = 1868) (out= 805)(deflated
56%)
adding: edu/hu/bigdata/InverterNewAPI$MapClass.class(in = 1274) (out= 486)(deflated
61%)
```

# I run the InverterNewAPI program in hadoop. The input is the patents data and
# the output is a folder named 152.

```
[joe@localhost inversion]$ hadoop jar wordcount.jar edu.hu.bigdata.InverterNewAPI
patents 152
16/02/20 10:52:53 INFO client.RMProxy: Connecting to ResourceManager at
/0.0.0.0:8032
16/02/20 10:52:57 INFO input.FileInputFormat: Total input paths to process : 1
16/02/20 10:52:58 INFO mapreduce.JobSubmitter: number of splits:2
16/02/20 10:52:58 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1455992537384_0003
16/02/20 10:52:59 INFO impl.YarnClientImpl: Submitted application
application_1455992537384_0003
16/02/20 10:52:59 INFO mapreduce.Job: The url to track the job:
http://localhost:8088/proxy/application_1455992537384_0003/
16/02/20 10:52:59 INFO mapreduce.Job: Running job: job_1455992537384_0003
16/02/20 10:53:12 INFO mapreduce.Job: Job job_1455992537384_0003 running in uber
mode : false
16/02/20 10:53:12 INFO mapreduce.Job:  map 0% reduce 0%
16/02/20 10:53:33 INFO mapreduce.Job:  map 8% reduce 0%
16/02/20 10:53:36 INFO mapreduce.Job:  map 22% reduce 0%
16/02/20 10:53:40 INFO mapreduce.Job:  map 26% reduce 0%
16/02/20 10:54:05 INFO mapreduce.Job:  map 40% reduce 0%
16/02/20 10:54:08 INFO mapreduce.Job:  map 47% reduce 0%
16/02/20 10:54:11 INFO mapreduce.Job:  map 48% reduce 0%
16/02/20 10:54:29 INFO mapreduce.Job:  map 56% reduce 0%
16/02/20 10:54:32 INFO mapreduce.Job:  map 67% reduce 0%
16/02/20 10:54:54 INFO mapreduce.Job:  map 68% reduce 0%
16/02/20 10:54:57 INFO mapreduce.Job:  map 71% reduce 0%
16/02/20 10:55:00 INFO mapreduce.Job:  map 75% reduce 0%
16/02/20 10:55:03 INFO mapreduce.Job:  map 78% reduce 0%
16/02/20 10:55:06 INFO mapreduce.Job:  map 81% reduce 0%
16/02/20 10:55:09 INFO mapreduce.Job:  map 84% reduce 0%
16/02/20 10:55:12 INFO mapreduce.Job:  map 88% reduce 0%
16/02/20 10:55:15 INFO mapreduce.Job:  map 91% reduce 0%
16/02/20 10:55:18 INFO mapreduce.Job:  map 94% reduce 0%
16/02/20 10:55:28 INFO mapreduce.Job:  map 97% reduce 0%
16/02/20 10:55:31 INFO mapreduce.Job:  map 100% reduce 0%
16/02/20 10:56:15 INFO mapreduce.Job:  map 100% reduce 17%
16/02/20 10:56:21 INFO mapreduce.Job:  map 100% reduce 67%
16/02/20 10:56:24 INFO mapreduce.Job:  map 100% reduce 70%
16/02/20 10:56:27 INFO mapreduce.Job:  map 100% reduce 73%
16/02/20 10:56:30 INFO mapreduce.Job:  map 100% reduce 76%
16/02/20 10:56:33 INFO mapreduce.Job:  map 100% reduce 79%
16/02/20 10:56:36 INFO mapreduce.Job:  map 100% reduce 82%
16/02/20 10:56:39 INFO mapreduce.Job:  map 100% reduce 84%
16/02/20 10:56:42 INFO mapreduce.Job:  map 100% reduce 87%
16/02/20 10:56:45 INFO mapreduce.Job:  map 100% reduce 90%
```

```
16/02/20 10:56:49 INFO mapreduce.Job:  map 100% reduce 92%
16/02/20 10:56:52 INFO mapreduce.Job:  map 100% reduce 94%
16/02/20 10:56:55 INFO mapreduce.Job:  map 100% reduce 97%
16/02/20 10:56:58 INFO mapreduce.Job:  map 100% reduce 100%
16/02/20 10:57:00 INFO mapreduce.Job: Job job_1455992537384_0003 completed
successfully
16/02/20 10:57:00 INFO mapreduce.Job: Counters: 49
	File System Counters
		FILE: Number of bytes read=594240678
		FILE: Number of bytes written=894018537
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=264079761
		HDFS: Number of bytes written=158078539
		HDFS: Number of read operations=9
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
	Job Counters
		Launched map tasks=2
		Launched reduce tasks=1
		Data-local map tasks=2
		Total time spent by all maps in occupied slots (ms)=270943
		Total time spent by all reduces in occupied slots (ms)=86521
		Total time spent by all map tasks (ms)=270943
		Total time spent by all reduce tasks (ms)=86521
		Total vcore-seconds taken by all map tasks=270943
		Total vcore-seconds taken by all reduce tasks=86521
		Total megabyte-seconds taken by all map tasks=277445632
		Total megabyte-seconds taken by all reduce tasks=88597504
	Map-Reduce Framework
		Map input records=16522439
		Map output records=16522439
		Map output bytes=264075431
		Map output materialized bytes=297120321
		Input split bytes=234
		Combine input records=0
		Combine output records=0
		Reduce input groups=3258984
		Reduce shuffle bytes=297120321
		Reduce input records=16522439
		Reduce output records=3258984
		Spilled Records=49567317
		Shuffled Maps =2
		Failed Shuffles=0
```

```
                Merged Map outputs=2
                GC time elapsed (ms)=5097
                CPU time spent (ms)=140720
                Physical memory (bytes) snapshot=500285440
                Virtual memory (bytes) snapshot=8162627584
                Total committed heap usage (bytes)=301146112
        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=264079527
        File Output Format Counters
                Bytes Written=158078539
```

# I show the **InverterNewAPI** and **Inverter** programs both produce the same result in
# the section with problem #5 results. This is located at the end of this document,
# linked here.

If you have your own working VM with installed CDH5.5.1, do this assignment on that VM. If, for what ever reason, you do not posses a working VM with CDH5.5.1, please be free to download Clouder's Getting started VM and do you assignment on that VM.

Capture all steps of your implementation with comments indicating what is it you are accomplishing with every step in an MS Word document. Upload to the class site. Please upload your working Java files as well. Please post comments and questions to the class Discussion Board on the Canvas site.

APPENDIX

Problem 1. Results

Problem 1, First 20 Results

```
"Come	1
"Defects,"	1
"I	1
"Information	1
"J"	1
"Plain	2
"Project	5
"Right	1
"Viator"	1
#4300]	1
$5,000)	1
&c,	2
&c.	1
'46.	1
'92	1
'AS-IS'	1
'Slife,	1
'TWAS	1
'Tis	8
'Tis,	1
```

Problem 2. Results
Problem 2, First 20 Results:

```
6542	and
3035	his
2712	he
1505	her
1362	you
1113	him
1042	all
908	He
769	she
768	they
766	had
760	out
737	not
708	my
699	Mr
677	their
661	up
649	like
640	me
617	have
```

Problem 3 Results

Problem 3, First 20 Results
Note: Format is Occurences, Number of words that occur that many times

1	32735
2	7065
3	2932
4	1647
5	1102
6	727
7	509
8	409
9	337
10	261
11	190
12	165
13	130
14	124
15	112
16	106
17	83
18	85
19	64
20	65

Problem 4 Results
Problem 4, First 20 Results

1	32735
2	7065
3	2932
4	1647
5	1102
6	727
7	509
8	409
9	337
10	261
11	190
12	165
13	130
14	124
15	112
16	106

```
17	83
18	85
19	64
20	65
```

Problem 5 Results

First 20 Results from InverterNewAPI.java

```
"CITED"	"CITING"
1	3964859,4647229
10000	4539112
100000		5031388
1000006		4714284
1000007		4766693
1000011		5033339
1000017		3908629
1000026		4043055
1000033		4190903,4975983
1000043		4091523
1000044		4082383,4055371
1000045		4290571
1000046		5525001,5918892
1000049		5996916
1000051		4541310
1000054		4946631
1000065		4748968
1000067		5071294,4944640,5312208
1000070		4928425,5009029
1000073		4107819,5474494
```

20 Results from Inverter.java on the patents data

```
"CITED"	"CITING"
1	3964859,4647229
10000		4539112
100000		5031388
1000006		4714284
1000007		4766693
1000011		5033339
1000017		3908629
1000026		4043055
1000033		4190903,4975983
1000043		4091523
1000044		4082383,4055371
1000045		4290571
```

```
1000046         5525001,5918892
1000049         5996916
1000051         4541310
1000054         4946631
1000065         4748968
1000067         5071294,4944640,5312208
1000070         4928425,5009029
1000073         4107819,5474494
```

The previous two results, and the entire list of results, match proving that InverterNewAPI is updated to the new API correctly.

Additionally, below are the last 10 results of each to further show the results for the new and old API match.

InverterNewAPI, last 10 results

```
999961
5782495,5738381,5878901,4171117,4262874,5048788,4871140,4832301,4437639
999965          5052613
999968          3916735
999971          3965843
999972          4038129
999973          4900344,5427610
999974          5464105,4560073,4728158
999977          4092587
999978          3915443
999983          5143114,5394715,5806555
```

Inverter, last 10 results, which match the above results

```
999961
5782495,5738381,5878901,4171117,4262874,5048788,4871140,4832301,4437639
999965          5052613
999968          3916735
999971          3965843
999972          4038129
999973          4900344,5427610
999974          5464105,4560073,4728158
999977          4092587
999978          3915443
999983          5143114,5394715,5806555
```