CS1003 W11 Practical 160026526 Tutor: Ruth Letham 25 April 2018

• Overview:
The aim of the practical was to create a program that implements a Map-Reduce to extract particular information from a small subset of Tweets made in January 2017 in JSON format by counting the occurrences of the "expended_url" links that Twitter records included in the bodies of users Tweets using Apache Hadoop.

• Design:
Examining the Twitter Api
To begin the practical, the formats of the Tweets are first studied to ensure that the program follows the specification. It can be seen that the Tweets are objects that contains three child objects which are user, entities and extended_entities. We require the urls which are located in the entities object which are stored in an array. In the array of urls, the expended_urls can be located and this is what the program will be counting.

Importing the library
The Hadoop library code is placed into the practical file as it is required when implementing Hadoop. IntelliJ is then configured to use Hadoop.

Writing the ScanWordsMapper method
The ScanWordsMapper class is written to extend the Mapper class. The Mapper class follows the Hadoop Map-Reduce framework in which it takes a key and value input and produces a key and value output. The inputs are received when calling the method and will be transformed into the outputs which will be sent to the Reducer to be partitioned. A method called map is then created that will take a LongWritable which will be the key input, a Text which will be the value input and a Context which will be the output, while throwing exceptions. The text value is a line in the json file which is first converted to a String called line. This will then allow it to be read by a new JsonReader called reader. The main Json object are the Tweets and so it is extracted into a tweetObject called tweetObject. The child object called entities is then created from the tweetObjects. The method then checks if the entities is not null, this is necessary as empty entities won't contain the expended url that is required. A JsonArray is then constructed called urls which is used to store the urls from the entities object. After checking that the urls are not null, a for loop is created which will loop around the urls array and the objects in the array are placed in items. The JsonValue of the "expended_url" are then gathered from the items object and an if statement checks to ensure the jsonValues are a String and not Null. The jsonValues are then converted to Strings which contain the expended urls. An output is then written which takes expendedUrls as its key and the 1 as the value outputs. Text is the output key as it is unique unlike the value output which are all 1. The output in this case is the maps from the expendedUrls key to the value 1.

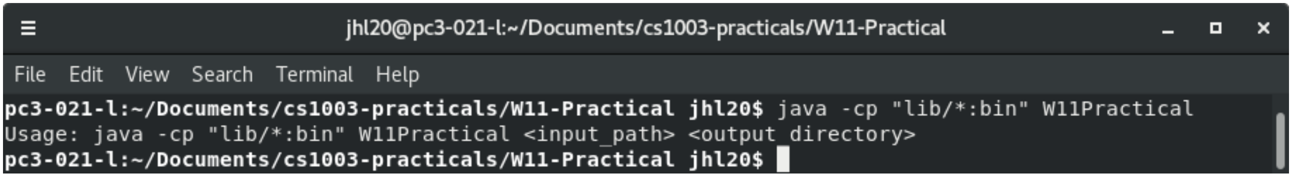
Writing the CountWordsReducer method
The CountWordsReducer class extends the Reducer class which takes a key and value input and transforms them into outputs. The reduce method will take the expendedUrls keys and the values 1 and the method will sum up the values for each unique url and will them write an output which maps the unique expendedUrls key with the total count value.

Writing the main method

In the main method, it first checks if the arguments length is less than 2 which prints out the usage and exits the program. The Strings input_path and output_path are then initialised as args[0] and args[1] respectively. A Configuration object is created and a Job object which has the name “Word Count” without a Cluster which can be created from the configuration object when necessary. The Path of the input directory for the map-reduce is then set to input_path and the Path of the output directory for the map-reduce is set to output_path. The ScanWordsMapper class is set as the Mapper class for job and the output key and values are set as a Text class and LongWritable class respectively which are words with count of 1. The CountWordsReducer is set as the Reducer class for job and the output key and values are set as a Text and LongWritable class which are the words with total counts. A try catch loop is made to submit the job to the cluster and wait for it to finish and catch any exceptions.

- Testing/Examples:

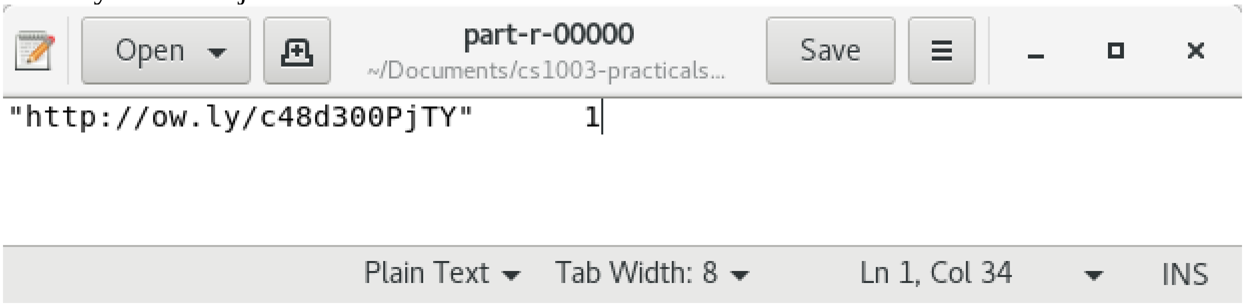
The program is first manually tested. The first test checks if the usage shows up when proper arguments are not present.



```
pc3-021-l:~/Documents/cs1003-practicals/W11-Practical jhl20$ java -cp "lib/*:bin" W11Practical
Usage: java -cp "lib/*:bin" W11Practical <input_path> <output directory>
pc3-021-l:~/Documents/cs1003-practicals/W11-Practical jhl20$
```

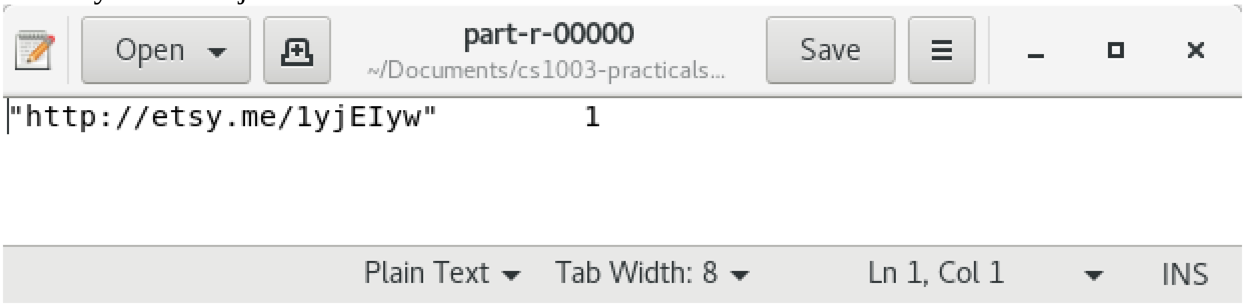
The program is then tested with /data-very-small/00.json, /data-very-small/01.json, /data-very-small and /1_minute. The text file outputs for the first three tests showed below and the fourth test shows the console output and a portion of the text file output.

/data-very-small/00.json :



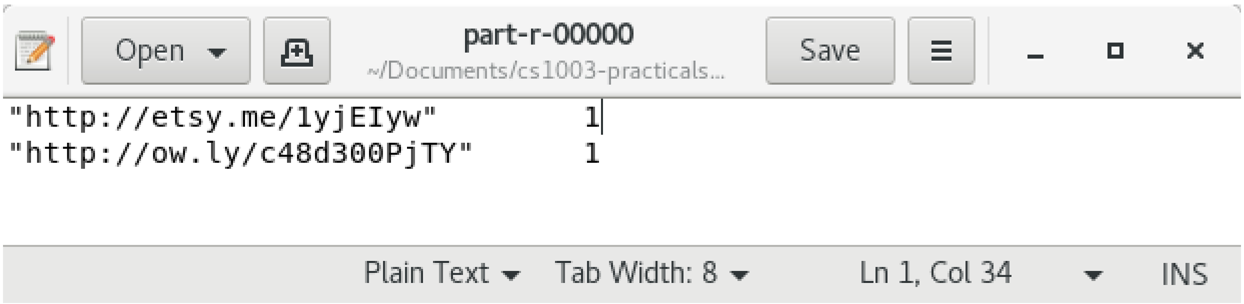
```
"http://ow.ly/c48d300PjTY"	1
```

/data-very-small/01.json :



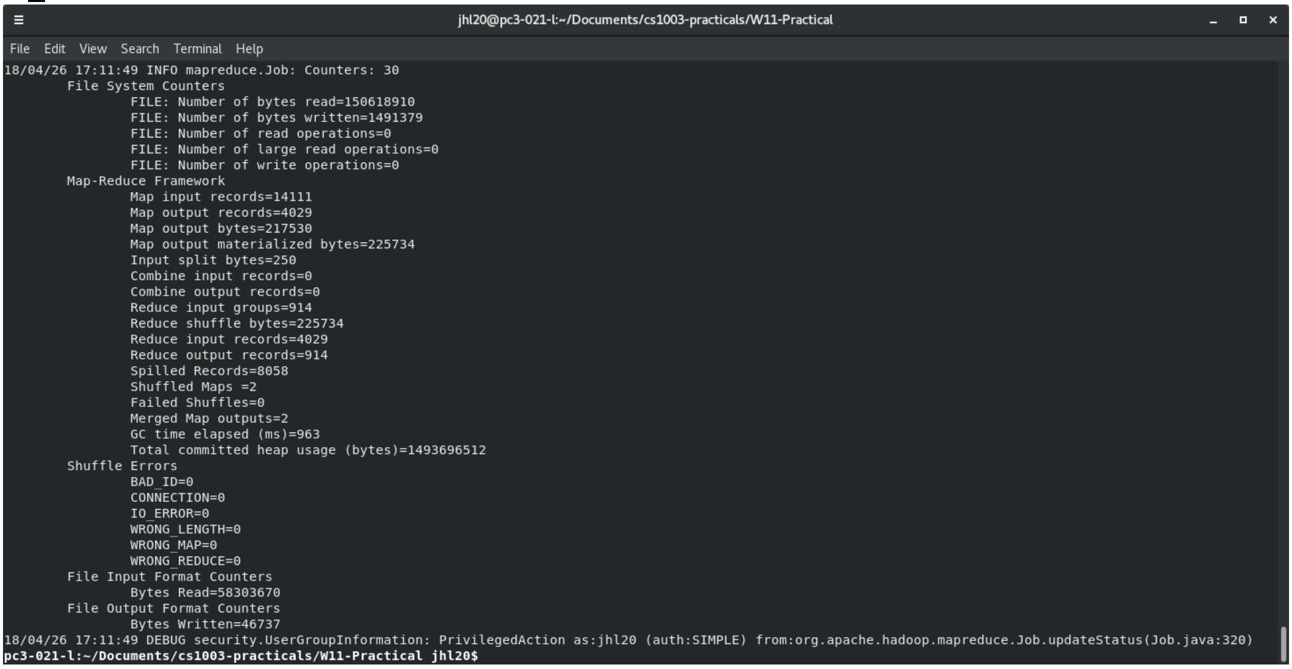
```
"http://etsy.me/1yjEIyw"	1
```

/data-very-small :

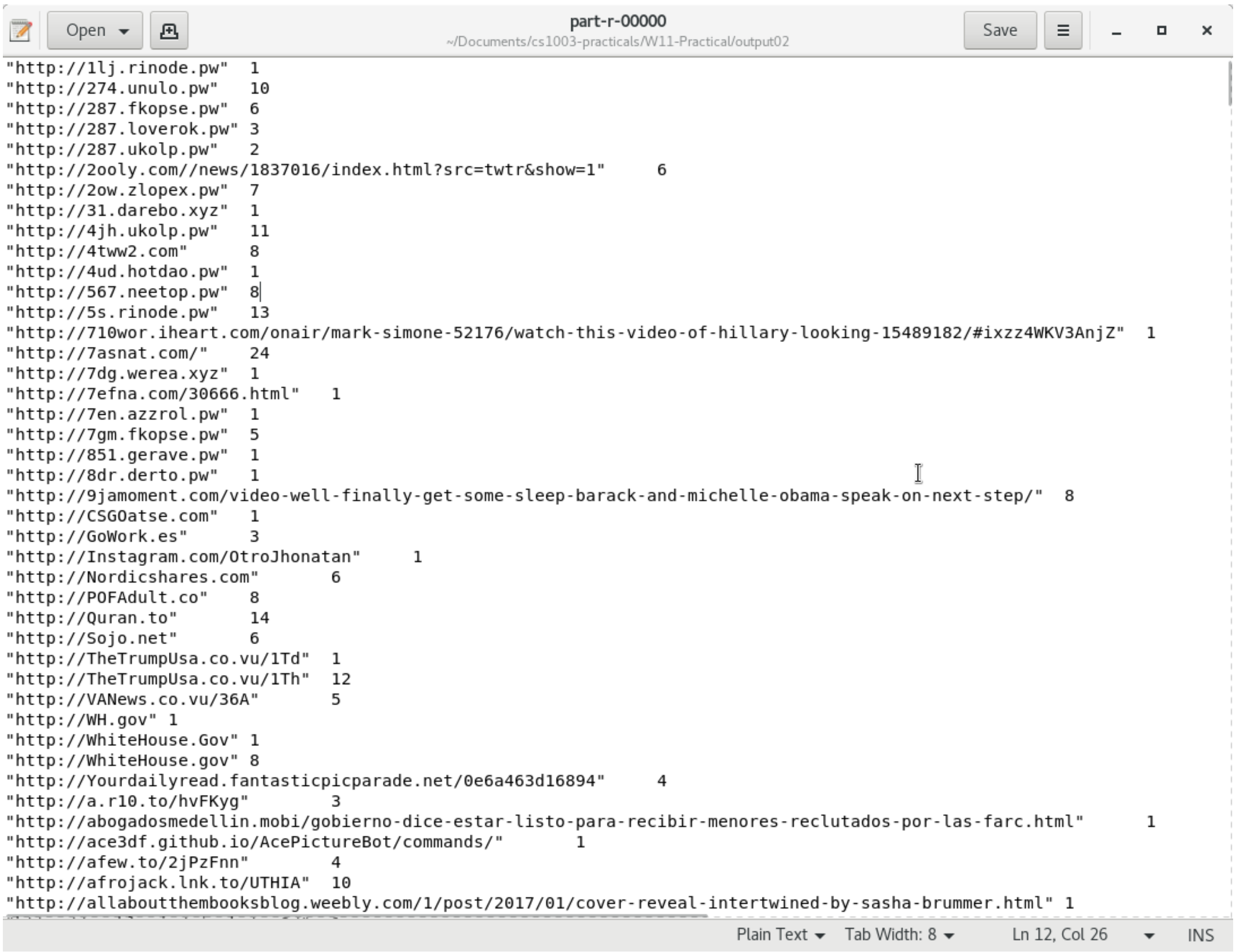


```
"http://etsy.me/1yjEIyw"	1
"http://ow.ly/c48d300PjTY"	1
```

/1_minute :

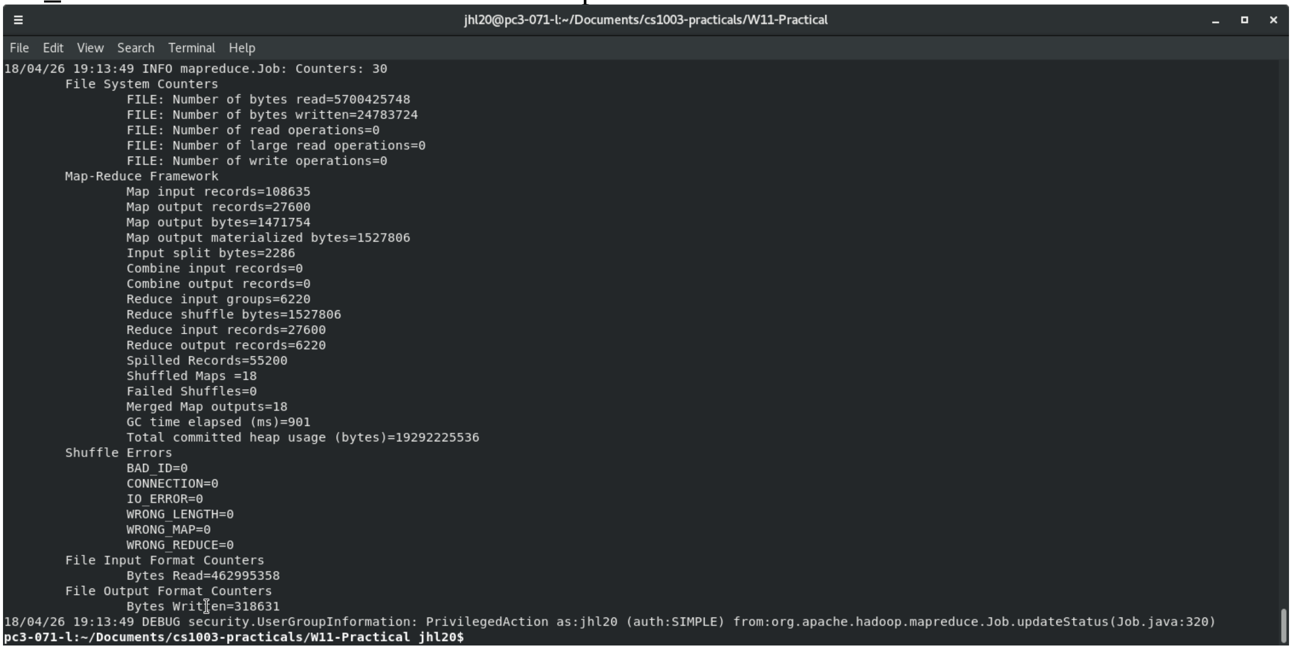

jhl20@pc3-021-l:~/Documents/cs1003-practicals/W11-Practical
File Edit View Search Terminal Help

```
18/04/26 17:11:49 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=150618910
                FILE: Number of bytes written=1491379
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=14111
                Map output records=4029
                Map output bytes=217530
                Map output materialized bytes=225734
                Input split bytes=250
                Combine input records=0
                Combine output records=0
                Reduce input groups=914
                Reduce shuffle bytes=225734
                Reduce input records=4029
                Reduce output records=914
                Spilled Records=8058
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=963
                Total committed heap usage (bytes)=1493696512
        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=58303670
        File Output Format Counters
                Bytes Written=46737
18/04/26 17:11:49 DEBUG security.UserGroupInformation: PrivilegedAction as:jhl20 (auth:SIMPLE) from:org.apache.hadoop.mapreduce.Job.updateStatus(Job.java:320)
pc3-021-l:~/Documents/cs1003-practicals/W11-Practical jhl20$
```



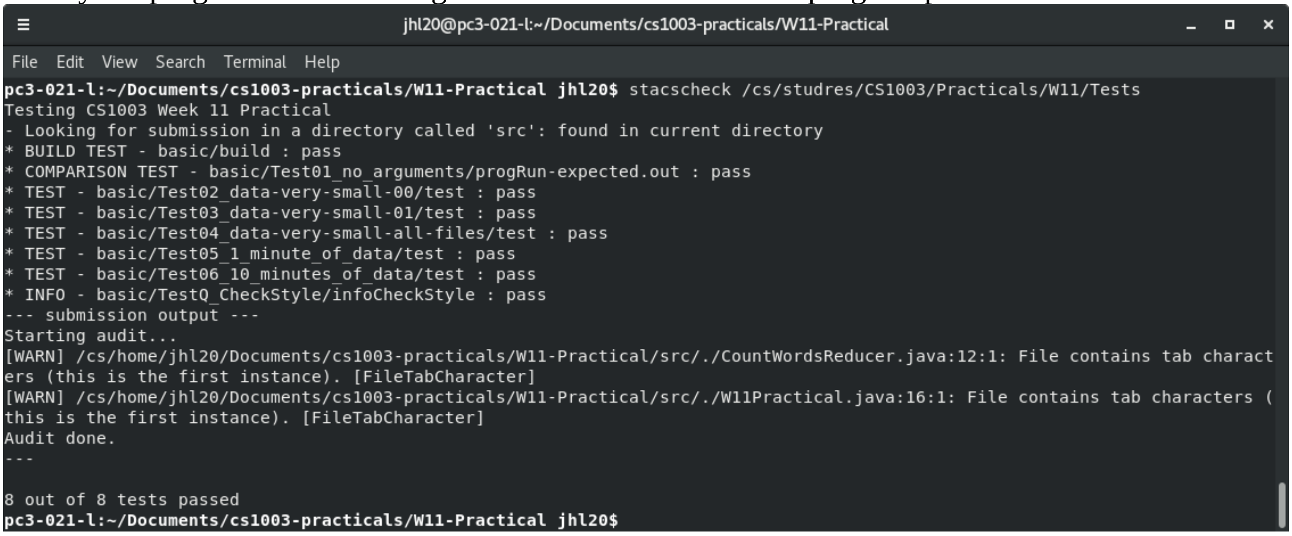
```
"http://1lj.rinode.pw"  1
"http://274.unulo.pw"   10
"http://287.fkopse.pw"  6
"http://287.loverok.pw" 3
"http://287.ukolp.pw"   2
"http://2ooly.com//news/1837016/index.html?src=twtr&show=1"     6
"http://2ow.zlopex.pw"  7
"http://31.darebo.xyz"  1
"http://4jh.ukolp.pw"   11
"http://4tww2.com"      8
"http://4ud.hotdao.pw"  1
"http://567.neetop.pw"  8
"http://5s.rinode.pw"   13
"http://710wor.iheart.com/onair/mark-simone-52176/watch-this-video-of-hillary-looking-15489182/#ixzz4WKV3AnjZ"  1
"http://7asnat.com/"    24
"http://7dg.werea.xyz"  1
"http://7efna.com/30666.html"   1
"http://7en.azzrol.pw"  1
"http://7gm.fkopse.pw"  5
"http://851.gerave.pw"  1
"http://8dr.derto.pw"   1
"http://9jamoment.com/video-well-finally-get-some-sleep-barack-and-michelle-obama-speak-on-next-step/"  8
"http://CSGOatse.com"   1
"http://GoWork.es"      3
"http://Instagram.com/OtroJhonatan"     1
"http://Nordicshares.com"       6
"http://POFAdult.co"    8
"http://Quran.to"       14
"http://Sojo.net"       6
"http://TheTrumpUsa.co.vu/1Td"  1
"http://TheTrumpUsa.co.vu/1Th"  12
"http://VANews.co.vu/36A"       5
"http://WH.gov" 1
"http://WhiteHouse.Gov" 1
"http://WhiteHouse.gov" 8
"http://Yourdailyread.fantasticpicparade.net/0e6a463d16894"     4
"http://a.r10.to/hvFKyg"        3
"http://abogadosmedellin.mobi/gobierno-dice-estar-listo-para-recibir-menores-reclutados-por-las-farc.html"      1
"http://ace3df.github.io/AcePictureBot/commands/"       1
"http://afew.to/2jPzFnn"        4
"http://afrojack.lnk.to/UTHIA"  10
"http://allaboutthembooksblog.weebly.com/1/post/2017/01/cover-reveal-intertwined-by-sasha-brummer.html" 1
```

/10_minutes was also tested. The console output is shown below.


jhl20@pc3-071-l:~/Documents/cs1003-practicals/W11-Practical
File Edit View Search Terminal Help

```
18/04/26 19:13:49 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=5700425748
                FILE: Number of bytes written=24783724
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=108635
                Map output records=27600
                Map output bytes=1471754
                Map output materialized bytes=1527806
                Input split bytes=2286
                Combine input records=0
                Combine output records=0
                Reduce input groups=6220
                Reduce shuffle bytes=1527806
                Reduce input records=27600
                Reduce output records=6220
                Spilled Records=55200
                Shuffled Maps =18
                Failed Shuffles=0
                Merged Map outputs=18
                GC time elapsed (ms)=901
                Total committed heap usage (bytes)=19292225536
        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=462995358
        File Output Format Counters
                Bytes Written=318631
18/04/26 19:13:49 DEBUG security.UserGroupInformation: PrivilegedAction as:jhl20 (auth:SIMPLE) from:org.apache.hadoop.mapreduce.Job.updateStatus(Job.java:320)
pc3-071-l:~/Documents/cs1003-practicals/W11-Practical jhl20$
```

Finally the program is tested using the statscheck in which the program passes all 8 tests.


jhl20@pc3-021-l:~/Documents/cs1003-practicals/W11-Practical
File Edit View Search Terminal Help

```
pc3-021-l:~/Documents/cs1003-practicals/W11-Practical jhl20$ stacscheck /cs/studres/CS1003/Practicals/W11/Tests
Testing CS1003 Week 11 Practical
- Looking for submission in a directory called 'src': found in current directory
* BUILD TEST - basic/build : pass
* COMPARISON TEST - basic/Test01_no_arguments/progRun-expected.out : pass
* TEST - basic/Test02_data-very-small-00/test : pass
* TEST - basic/Test03_data-very-small-01/test : pass
* TEST - basic/Test04_data-very-small-all-files/test : pass
* TEST - basic/Test05_1_minute_of_data/test : pass
* TEST - basic/Test06_10_minutes_of_data/test : pass
* INFO - basic/TestQ_CheckStyle/infoCheckStyle : pass
--- submission output ---
Starting audit...
[WARN] /cs/home/jhl20/Documents/cs1003-practicals/W11-Practical/src/./CountWordsReducer.java:12:1: File contains tab charact
ers (this is the first instance). [FileTabCharacter]
[WARN] /cs/home/jhl20/Documents/cs1003-practicals/W11-Practical/src/./W11Practical.java:16:1: File contains tab characters (
this is the first instance). [FileTabCharacter]
Audit done.
---

8 out of 8 tests passed
pc3-021-l:~/Documents/cs1003-practicals/W11-Practical jhl20$
```

• Evaluation:

In this practical, the program was required to implement a Map-Reduce on a small amount of Tweets to extract particular information by counting the occurrences of the “expended_url” links that Twitter records included in the bodies of users Tweets using Apache Hadoop. The program was able to perform the task as required from the specification. The program creates a new file output which contains two text files, an empty one called “_SUCCESS” and one that contains the output of the Map-Reduce called “part-r-00000”. Other than that, the program ran the autocheckers without encountering any problems. Since the program carried out the requirements set by the practical therefore it can be said to be successful.

- Conclusion:

In conclusion, the program was successful in using Map-Reduce to gather the unique “extended urls” and the total values while using Apache Hadoop. The practical seemed difficult at the beginning as it was the first practical that required knowledge of JSON and Hadoop. After reading up on the documentation, the practical was easier to carry out. The example code from the lectures also provided a baseline to start of the practical which allowed for cleaner implementation of code. Extensions were not attempted and the program will only perform the initial task provided.