
PROGRAMMING ASSIGNMENT 2: MAPREDUCE ON CLOUDLAB

COSC560

April 6, 2019

Clara Nguyen

Rachel Offutt

University of Tennessee EECS

Contents

0.1	Project Summary	2
0.2	Project Specific Requirements	2
0.2.1	Identifying and removing stop words	2
0.2.2	Building the Inverted Index	2
0.2.3	Query the Inverted Index	3
0.3	Project Design Choices	3
0.3.1	Language Choice	3
0.3.2	Configuration	3
0.3.3	Setup of Output File	3
0.3.4	Stop Words	4
0.4	Running the Project	4
0.5	Implementation Screenshots	5

0.1 PROJECT SUMMARY

In this assignment, you will program the Map Reduce parallel data processing system on the Cloudlab cloud computing platform. This will allow you gain practical experience on Map Reduce programming and learn the performance implications of parallel data processing. The following basic goals must be fulfilled:

- Cloudlab setup
- Create Cloudlab cluster on Hadoop
- Build the MapReduce code for the Reverse-indexer
- Run the MapReduce program on the Hadoop cluster
- Query the inverted index

0.2 PROJECT SPECIFIC REQUIREMENTS

0.2.1 Identifying and removing stop words

One issue is that some words are so common that their presence in an inverted index is "noise," that is they can obfuscate the more interesting properties of a document. Such words are called "stop words." For this part of the assignment, write a word count Map Reduce function to perform a word count over a corpus of text files and to identify stop words. It is up to you to choose a reasonable threshold (word count frequency) for stop words, but make sure you provide adequate justification and explanation of your choice. A parser will group words by attributes which are not relevant to their meaning (e.g., "hello", "Hello", and "HELLO" are all the same word), so it is up to you to define "scrub" however you wish; some suggestions include case-insensitivity, etc. It is not required that you treat "run" and "ran" as the same word, but your parser should handle case insensitivity. Once you have written your code, then run your code and collect the word counts for submission with all your Mapper and Reducer files.

0.2.2 Building the Inverted Index

For this portion of the assignment, you will design a MapReduce-based algorithm to calculate the inverted index. To this end, you are to create a full inverted index, which maps

words to their documentID + line number in the document. Note that your final inverted index should not contain the words identified in Step 1. The format of your MapReduce output (i.e., the inverted index) must be simple enough to be machine-parseable; it is not impossible to imagine your index being one of many data structures used in a search engine's indexing pipeline. Your submitted indexer should be able to run successfully on one or multiple input txt files, where "successfully" means it should run to completion without errors or exceptions, and generate the correct word->DocID mapping. You are required to submit all relevant Mapper and Reducer Java files, in addition to any supporting code or utilities.

0.2.3 Query the Inverted Index

Write a query program on top of your full inverted file index that accepts a user-specified query (one or more words) and returns not only the documentIDs but also the locations in the form of line numbers. The query program can be local: it does not need to handle the task using Map-Reduce framework again. It is not required that your query program to return text snippets from the original text files.

0.3 PROJECT DESIGN CHOICES

0.3.1 Language Choice

We chose to implement this project in Python because data parsing, especially word parsing, is exponentially easier in Python and one of our team members has extensive experience in working with big data and text parsing in Python.

0.3.2 Configuration

We created a script file to run the mapReduce functions to allow for ease of use for the user. The user provides a list of input arguments in the form of file names they would like to run on.

0.3.3 Setup of Output File

The output file is set up in terms of word, then file it appears in, then the lines it appears on. We structured our output file this way to make querying easier.

0.3.4 Stop Words

To parse our data, we see if there are over 1000 instances of a word and if the word is five letters or less. The reasoning behind our design decision is because when using the complete works of Shakespeare, we opened up the total count of words, and found that generally words that appeared more than 1000 times were likely to be stop words, and the 5 character cutoff is to ensure that the majority of the names are not cut off as well.

0.4 RUNNING THE PROJECT

NOTE: YOU MUST HAVE PYTHON3 ENABLED IN ORDER TO RUN THE QUERYING SCRIPT. PLEASE TYPE THE FOLLOWING COMMAND:

```
source /opt/rh/python33/enable
```

1. Step by step running

- Open up a terminal in Linux with the project downloaded.
- Change your working directory to

```
/mapreduce/
```

- Type the following command:

```
sh execute.sh <list of files to run on>
```

- After MapReduce finishes, the querying script will run. You can now enter in a specific word or list of words to search for. (See the following section for screenshots of code working and example use cases.)

2. screenshot of run command

A terminal window screenshot showing a user prompt 'iDestyKK@namenode: ~/project_thing\$' followed by the command './execute.sh data/100-0.txt data/test.txt'.

Figure 1: An example of the run script.

0.5 IMPLEMENTATION SCREENSHOTS

NOTE: We have already demoed to the TA for this course and have proven that our implementation of the MapReduce on Hadoop is correct and in working order.

```
iDestyKK@namenode:~/project_thing$ ./execute.sh data/100-0.txt
Deleted /exp_data
19/04/05 11:02:07 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [py/mapper.py, py/reducer.py, /tmp/hadoop-unjar6916433100988601871/] [] /tmp/streamjob6051483506652354709.jar tmpDir=null
19/04/05 11:02:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:02:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:02:08 INFO mapred.FileInputFormat: Total input paths to process : 1
19/04/05 11:02:08 INFO mapreduce.JobSubmitter: number of splits:2
19/04/05 11:02:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1554481564388_0002
19/04/05 11:02:08 INFO impl.YarnClientImpl: Submitted application application_1554481564388_0002
19/04/05 11:02:08 INFO mapreduce.Job: The url to track the job: http://resourcemanager.thetest.educationproject-pg0.wisc.cloudlab.us:8088/proxy/application_1554481564388_0002/
19/04/05 11:02:08 INFO mapreduce.Job: Running job: job_1554481564388_0002
19/04/05 11:02:15 INFO mapreduce.Job: Job job_1554481564388_0002 running in uber mode : false
19/04/05 11:02:15 INFO mapreduce.Job: map 0% reduce 0%
19/04/05 11:02:24 INFO mapreduce.Job: map 100% reduce 0%
19/04/05 11:02:33 INFO mapreduce.Job: map 100% reduce 100%
19/04/05 11:02:33 INFO mapreduce.Job: Job job_1554481564388_0002 completed successfully
```

Figure 2: Proof of program running with one text input file

```
iDestyKK@namenode:~/project_thing$ ./execute.sh data/100-0.txt data/test.txt
Deleted /exp_result
Deleted /exp_data
19/04/05 11:09:26 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [py/mapper.py, py/reducer.py, /tmp/hadoop-unjar8064231634160701378/] [] /tmp/streamjob6456627397334723977.jar tmpDir=null
19/04/05 11:09:27 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:09:27 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:09:28 INFO mapred.FileInputFormat: Total input paths to process : 2
19/04/05 11:09:28 INFO mapreduce.JobSubmitter: number of splits:3
19/04/05 11:09:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1554481564388_0004
19/04/05 11:09:28 INFO impl.YarnClientImpl: Submitted application application_1554481564388_0004
19/04/05 11:09:28 INFO mapreduce.Job: The url to track the job: http://resourcemanager.thetest.educationproject-pg0.wisc.cloudlab.us:8088/proxy/application_1554481564388_0004/
19/04/05 11:09:28 INFO mapreduce.Job: Running job: job_1554481564388_0004
19/04/05 11:09:34 INFO mapreduce.Job: Job job_1554481564388_0004 running in uber mode : false
19/04/05 11:09:34 INFO mapreduce.Job: map 0% reduce 0%
19/04/05 11:09:38 INFO mapreduce.Job: map 33% reduce 0%
19/04/05 11:09:40 INFO mapreduce.Job: map 100% reduce 0%
19/04/05 11:09:52 INFO mapreduce.Job: map 100% reduce 100%
19/04/05 11:09:52 INFO mapreduce.Job: Job job_1554481564388_0004 completed successfully
```

Figure 3: Proof of program running with multiple text input files

```

Deleted /exp_data
19/04/05 11:02:07 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [py/mapper.py, py/reducer.py, /tmp/hadoop-unjar6916433100988601871/] [] /tmp/streamjob6051483506652354709.jar tmpDir=null
19/04/05 11:02:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:02:07 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/10.10.1.2:8032
19/04/05 11:02:08 INFO mapred.FileInputFormat: Total input paths to process : 1
19/04/05 11:02:08 INFO mapreduce.JobSubmitter: number of splits:2
19/04/05 11:02:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1554481564388_0002
19/04/05 11:02:08 INFO impl.YarnClientImpl: Submitted application application_1554481564388_0002
19/04/05 11:02:08 INFO mapreduce.Job: The url to track the job: http://resourcemanager.thetest.educationproject-pg0.wisc.cloudlab.us:8088/proxy/application_1554481564388_0002/
19/04/05 11:02:08 INFO mapreduce.Job: Running job: job_1554481564388_0002
19/04/05 11:02:15 INFO mapreduce.Job: Job job_1554481564388_0002 running in uber mode : false
19/04/05 11:02:15 INFO mapreduce.Job: map 0% reduce 0%
19/04/05 11:02:24 INFO mapreduce.Job: map 100% reduce 0%
19/04/05 11:02:33 INFO mapreduce.Job: map 100% reduce 100%
19/04/05 11:02:33 INFO mapreduce.Job: Job job_1554481564388_0002 completed successfully
19/04/05 11:02:33 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=51228201
  FILE: Number of bytes written=102833966
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=5826493
  HDFS: Number of bytes written=4003073
  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)=11564
  Total time spent by all reduces in occupied slots (ms)=6711
  Total time spent by all map tasks (ms)=11564
  Total time spent by all reduce tasks (ms)=6711
  Total vcore-milliseconds taken by all map tasks=11564
  Total vcore-milliseconds taken by all reduce tasks=6711
  Total megabyte-milliseconds taken by all map tasks=11841536
  Total megabyte-milliseconds taken by all reduce tasks=6872064

```

Figure 4: Proof of program completion

```

Enter word(s) to query:
youth clown hadoop zeal you
'youth' appears in the following documents and on the following lines:

hdfs://namenode:9000/exp_data/100-0.txt 13712 22888 23761 10724 26574 7615 12713 6200 12686 30461 39039 10899 21152 1212 521 23784 58893 2609 38835 3644 58901 $
0311 46483 13572 26613 13567 13556 38221 13546 13540 71179 3641 871 33744 72274 1852 1853 24501 13513 13507 53172 255 8416 44532 34575 39896 35836 4535 51639 25
677 50724 40105 24704 1447 3394 35759 3378 23176 72443 45658 22463 34446 9176 14453 32867 14447 38291 70209 12371 4624 30731 65978 40897 72587 45714 405 26762 $
7379 14313 36865 40243 403 27936 47422 14287 59541 10306 37391 14210 17246 1108 12320 25816 791 7256 2111 23190 27071 12310 23196 14175 45784 23223 6909 $
4114 12301 3764 46651 25856 25859 27243 325 72077 52528 23309 41293 23312 36855 13977 23344 1890 41298 28975 72078 16817 36857 38781 40784 50030 33818 49969 415
8 31797 12802 49904 49279 26262 13889 37888 13856 13848 13806 12745 39400 36171 30172 30173 36801 26333 30795 62562 68582 79966 78151 20047 42446 77795 13036 65
325 64430 44910 78138 54780 9748 77757 6103 13118 9424 64527 11164 20989 68595 77605 77606 67587 70404 66886 68480 78144 30813 62583 66546 79438 78136 78140 535
17 38796 66858 60328 56646 14707 6213 22675 69966 13692 69389 63130 30803 72724 65984 3808 3507 17389 68273 42464 13070 78632 54649 76423 78147 54202 21467 6455
3 43543 60470 57131 65112 78146 31391 63056 78149 78696 23487 57135 18636 64474 4092 4090 64976 42407 65039 64710 76126 65037 66099 68598 76175 36743 8475 51895
64539 77215 3136 39440 50476 67440 22459 61980 6178 61400 57647 68170 3114 37187 15940 13928 81855 8857 77710 66738 64375 78094 4159 64429 22873 73156 33925 55
599 22345 42405 54885 3141 64577 67669 77757 8154 36638 58858 30866 12153 11408 52763 75720 63509 70029 845 66552 5148 56675 78142 75769 66869 17388 52264 64585
54378 52075

'clown' appears in the following documents and on the following lines:

hdfs://namenode:9000/exp_data/100-0.txt 62639 62633 62626 64266 76269 75977 55059 62619 62612 64273 65655 23696 23703 65663 65669 23694 65675 65458 76068 65599
24344 23715 65684 65591 65422 65690 65580 65697 65705 76854 65574 65713 65721 65563 65729 75740 75744 65412 65737 65554 65744 65751 76871 65405 65403 23708 2435
0 23712 63751 65541 63744 65533 63738 76852 63732 65821 65826 65832 65628 23701 65843 65849 65855 65861 63701 65871 65877 24336 65884 65892 63690 55149 65902 65
908 77378 24334 65616 76880 24331 24329 76226 24326 23720 77341 66158 66173 66193 23722 76848 76895 63482 65434 63475 63469 63460 76803 66314 66320 63451 66329
66336 66343 66363 76906 63413 6672 63393 76813 76910 65639 63385 66456 63379 66485 23699 63354 63340 7105 75737 76805 55072 63331 63317 76917 74133 63303 63298
55056 76919 62088 76238 2824 76795 75781 5869 75913 75915 76834 75918 77335 77349 75775 77371 55064 55147 55090 55138 76829 75930 55079 75933 75989 75937 76706
75941 75943 55143 76800 76347 65454 75987 64191 75948 64196 75757 77356 65622 75769 55141 64202 75951 75985 75954 62854 62851 76246 76316 62843 64211 55084 6425
6 55062 62822 53016 22147 76249 65607 64219 62803 75959 77368 76305 55051 64225 62770 64233 62747 64240 75964 76821 64279 62736 76292 75761 62730 75981 62724 65
717 76289 62711 64248 62703 76285 75735 62689 64257 62682 77365 62673 76277 76193 62661 62654 75973 62647 77363 77347 76197 30365 30658 30387 30457 30357 6948 $
0428 4229 7041 6942 12840 11847 11841 30412 30643 3508 4246 14358 49572 4891 7164 4323 4869 30390 6954 2972 3500 3521 4317 4286 3583 71771 30579 30440 4221 3045
5 4292 3515 11844 4273 30616 27289 30508 30371 4216 30437 30397 7172 11859 11829 7913 14355 30622 7053 4875 5252 12678 30573 30567 4308 6967 30628 6930 3550 355
8 30377 7043 30418 4257 6904 30443 4280 4856 5254 30665 11833 30561 30544 4861 30434 7060 3486 7156 30406 3567 30393 4264 5261 4300 30368 6974 30589 3463 12777
70639 6936 30553 30596 30635 11850 4239 30446 4907 6981 5232 4898 30403 71968 3480 30380 6995 30451 30652 30421 30361 14320 5223 30603 6923 7179 30610 5213 6965
5207 4914 5268 71967 3535 11853 5201 30476 11823 30409 3494 3594

I'm sorry, 'hadoop' does not appear in any of the files.

'zeal' appears in the following documents and on the following lines:

hdfs://namenode:9000/exp_data/100-0.txt 58634 57615 55846 73088 35405 55156 52862 59415 38527 37758 57952 57861 73089 72110 54500 36193 48054 39639 40525 72208
60028 35872 30143 59849 32880 18151 35765 67422 51449 53517 9813 77069 52536

I'm sorry, 'you' does not appear in any of the files.

```

Figure 5: Query function showing line display for words in line, not in file, and the removal of stop words.