Advanced Topics in Databases
Assignment 1

Iraklis Bekiaris

October 22, 2017

General Notes:

A Class Settings has been implemented in order to pass arguments that refers to the desired options of the jar execution. For example if we want to execute a job with more than one reducers we can just pass the arguments -numReducers (x) and the execution will be done with x Reducers.

For Exercises 2a, 2b, 3 there are some options added. These options are:

- -combine true | where you set the program to run with a combiner
- -numReucers x | where you set the program to run with x reducers
- -compress true | where you set the program to run with compression
- -skip file | where you set the program to skip patterns you dont want to include in your results (stopwords for example)

For exercise2b.InvertedIndex.java:

- `-doc_to_count_words doc` | where you set the program to count the number of words the doc contains

Limitations

- exercise1.StopWords.java and exercise2a.StopWordsPerformance should have standard arguments /input /stopwords /topK
- exercise2b.InvertedIndex.java should run WITHOUT a combiner and have standard arguments /input `/inverted_index`
- exercise3.InvertedIndexExtention should run WITH a combiner (-combiner true) and have standard arguments /input `/inverted_index_extention`

Examples of running exercise2b.InvertedIndex.java

- hadoop jar /home/cloudera/project.jar exercise2b.InvertedIndex /input /output -combiner true -skip stopwords.csv
- hadoop jar /home/cloudera/project.jar exercise2b.InvertedIndex /input /output -skip stopwords.csv `-doc_to_count_words` 4

1. Exercise 1

For the purposes of the exercise, 2 Jobs has been used.

1st Job:

The Job is similar to WordCount problem. The mappers stor as Key a word and as Value 1. On the reduce phase, the reducers computes the sum of the values and stores as Key the word and as Value the sum.

We are interested only for the StopWords though and, as a result, we store only the Words where their sum is greater than 4000.

2nd Job:

So, on the 2nd Job we want to compute the topK StopWords. The reason we need the second Job is for sorting by value the output of the previous job. This can be done easily by switching the Key Value pairs where the Key will become Value and the Value Key and the sorting will be done this time with respect to the frequency of the word.

In order to have descending sorting, we need to implement an IntComparator and override the compare method. After that we set the sortComparatorClass to listen to our Comparator and in this way we have descending sorting of the Keys.

On the reduce phase where the input is now sorted by frequency we can take the topK StopWords with the help of a counter. Also, in the reduce phase we are creating the file stopwords.csv by using MultipleOutputs. After the second job finishes we rename the file stopwords.csv-r-00000 to stopwords.csv and we also move it to hdfs home directory. This is done automatically in the code, there is no need for after terminal commands.

Important: we cannot use more than 1 reducers on the second job because we are facing a sorting problem where we want all of our elements to be in one file and not splitted in order to have the topK. We could implement this to be done with more reducers but there is no point since no matter how many documents we have in the end we come with a small document because, how many words exist in the english language..

2. Exercise 2

2.1 Part a

Settings [1 Reducer, No Combiner, No Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:18:37 PDT 2017
Started:	Sun Oct 22 04:18:41 PDT 2017
Finished:	Sun Oct 22 04:19:09 PDT 2017
Elapsed:	27sec
Diagnostics:	
Average Map Time	13sec
Average Shuffle Time	6sec
Average Merge Time	1sec
Average Reduce Time	3sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:18:38 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	0	1

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:19:12 PDT 2017
Started:	Sun Oct 22 04:19:18 PDT 2017
Finished:	Sun Oct 22 04:19:30 PDT 2017
Elapsed:	11sec
Diagnostics:	
Average Map Time	3sec
Average Shuffle Time	2sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:19:15 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

Settings [10 Reducer, No Combiner, No Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:20:38 PDT 2017
Started:	Sun Oct 22 04:20:42 PDT 2017
Finished:	Sun Oct 22 04:21:28 PDT 2017
Elapsed:	46sec
Diagnostics:	
Average Map Time	14sec
Average Shuffle Time	8sec
Average Merge Time	0sec
Average Reduce Time	2sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:20:39 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	1	10

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:21:31 PDT 2017
Started:	Sun Oct 22 04:21:36 PDT 2017
Finished:	Sun Oct 22 04:22:02 PDT 2017
Elapsed:	25sec
Diagnostics:	
Average Map Time	8sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:21:33 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

Execution Time is 46+25secs = 71secs

Settings [10 Reducer, Combiner, No Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:22:43 PDT 2017
Started:	Sun Oct 22 04:22:48 PDT 2017
Finished:	Sun Oct 22 04:23:31 PDT 2017
Elapsed:	43sec
Diagnostics:	
Average Map Time	15sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:22:45 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	0	10

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:23:34 PDT 2017
Started:	Sun Oct 22 04:23:39 PDT 2017
Finished:	Sun Oct 22 04:24:04 PDT 2017
Elapsed:	25sec
Diagnostics:	
Average Map Time	8sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:23:36 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

Execution Time is 43+25secs = 68secs
The use of the Combiner gives a 3second faster execution of the process. This might be small here but for real big projects this could be a big difference.

Settings [10 Reducer, Combiner, Compression]

Job Overview

Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:24:48 PDT 2017
Started:	Sun Oct 22 04:24:53 PDT 2017
Finished:	Sun Oct 22 04:25:36 PDT 2017
Elapsed:	43sec
Diagnostics:	
Average Map Time	14sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	1sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:24:50 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	0	10

Job Overview

Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:25:37 PDT 2017
Started:	Sun Oct 22 04:25:42 PDT 2017
Finished:	Sun Oct 22 04:26:08 PDT 2017
Elapsed:	25sec
Diagnostics:	
Average Map Time	8sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:25:39 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

Execution Time is 43+25secs = 68secs
There is no difference in the execution time... We see difference in the use of compression in the case where we have 50 Reducers later.

Settings [10 Reducer, No Combiner, Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:27:05 PDT 2017
Started:	Sun Oct 22 04:27:10 PDT 2017
Finished:	Sun Oct 22 04:27:56 PDT 2017
Elapsed:	46sec
Diagnostics:	
Average Map Time	14sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	2sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:27:06 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	1	6
Reduces	0	1	10

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:27:59 PDT 2017
Started:	Sun Oct 22 04:28:05 PDT 2017
Finished:	Sun Oct 22 04:28:31 PDT 2017
Elapsed:	25sec
Diagnostics:	
Average Map Time	8sec
Average Shuffle Time	9sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:28:02 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	10	10
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	10
Reduces	0	0	1

Execution Time is 46+25secs = 71secs

Settings [50 Reducer, Combiner, Compression]

Job Overview

Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:47:15 PDT 2017
Started:	Sun Oct 22 04:47:20 PDT 2017
Finished:	Sun Oct 22 04:49:29 PDT 2017
Elapsed:	2mins, 8sec
Diagnostics:	
Average Map Time	15sec
Average Shuffle Time	10sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:47:17 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	50	50

Attempt Type	Failed	Killed	Successful
Maps	0	0	6
Reduces	0	0	50

Job Overview

Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:49:31 PDT 2017
Started:	Sun Oct 22 04:49:36 PDT 2017
Finished:	Sun Oct 22 04:51:23 PDT 2017
Elapsed:	1mins, 46sec
Diagnostics:	
Average Map Time	9sec
Average Shuffle Time	1mins, 17sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:49:33 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	50	50
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	50
Reduces	0	0	1

Execution Time is 2.08 + 1.46 mins = 3.54 mins

We notice that there is a big difference in the execution time. This is happenings because 6 mappers send data to 50 reducers and after that 50 reducers send data to 50 mappers where they send data to 1 reducer. That means that we have to many ios and more job in the shuffle and sort phase. For those 2 reasons we have to be careful on the number of reducers we choose.

Settings [50 Reducer, Combiner, No Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:54:18 PDT 2017
Started:	Sun Oct 22 04:54:23 PDT 2017
Finished:	Sun Oct 22 04:56:47 PDT 2017
Elapsed:	2mins, 23sec
Diagnostics:	
Average Map Time	16sec
Average Shuffle Time	10sec
Average Merge Time	0sec
Average Reduce Time	1sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:54:20 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	50	50

Attempt Type	Failed	Killed	Successful
Maps	0	0	6
Reduces	0	0	50

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 04:56:49 PDT 2017
Started:	Sun Oct 22 04:56:55 PDT 2017
Finished:	Sun Oct 22 04:58:43 PDT 2017
Elapsed:	1mins, 47sec
Diagnostics:	
Average Map Time	9sec
Average Shuffle Time	1mins, 16sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 04:56:51 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	50	50
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	50
Reduces	0	0	1

Execution Time is 2.23 + 1.47 mins = 4.10 mins

We notice that without the Combiner we have a 16 seconds slowest peroformance

Settings [50 Reducer, No Combiner, No Compression]

Job Overview	
Job Name:	stop_words
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 05:02:47 PDT 2017
Started:	Sun Oct 22 05:02:52 PDT 2017
Finished:	Sun Oct 22 05:05:22 PDT 2017
Elapsed:	2mins, 29sec
Diagnostics:	
Average Map Time	22sec
Average Shuffle Time	10sec
Average Merge Time	0sec
Average Reduce Time	1sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 05:02:49 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	50	50

Attempt Type	Failed	Killed	Successful
Maps	0	0	6
Reduces	0	0	50

Job Overview	
Job Name:	topK
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 05:05:24 PDT 2017
Started:	Sun Oct 22 05:05:31 PDT 2017
Finished:	Sun Oct 22 05:07:20 PDT 2017
Elapsed:	1mins, 49sec
Diagnostics:	
Average Map Time	9sec
Average Shuffle Time	1mins, 19sec
Average Merge Time	0sec
Average Reduce Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 05:05:26 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	50	50
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	50
Reduces	0	0	1

Execution Time is 2.29 + 1.49 mins = 4.18 mins

Conclusion

After all, we come to the conclusion that we have the best performance using only 1 Reducer. In general, for all the exercises using only 1 Reducer seems to be the best case. I understand that this is not logical and for real problems with big data this would not happen due to memory issues and ios but for this problem 1 Reducer is the best case according to the implementations.

2.2 Part b

1 Job has been used for the exercise

On the map phase we get the location and name of the file we map and we write to the context as Key the word and as Value the name of the file. We also skip the words from stopwords.csv we created in the exercise 1. These is implemented with the help of a HashSet where we add the words we want to skip.

On the reduce phase we get the files that a word has been seen and we put them all together as Value. Key is again the word. We also increase our counter every time a word comes from a specific file that we want to count how many words exists there.

After the job is finished, we get our counter and we save his value in a file on hdfs. This is happening programmatically and the file is saved on the hdfs home directory as counters.txt.

From the counters section on the job info we can find locally in http://quickstart.cloudera:19888/jobhistory/ we can determine the unique records which are shown by the counter 'Reduce output records' which is 56108

Note that this exercise is not implemented to run with a Combiner.

After experiments, the most efficient way to run the program is using a single Reducer as mentioned on the Conclusion of exercise 2 part a.

Job Overvie

Job Name:	inverted_index
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Oct 22 09:35:49 PDT 2017
Started:	Sun Oct 22 09:35:55 PDT 2017
Finished:	Sun Oct 22 09:36:28 PDT 2017
Elapsed:	33sec
Diagnostics:	
Average Map Time	17sec
Average Shuffle Time	10sec
Average Merge Time	1sec
Average Reduce Time	3sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Oct 22 09:35:51 PDT 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	6	6
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	6
Reduces	0	0	1

3. Exercise 3

1 Job has been used for the exercise

On the map phase we follow the same map implementation as in exercise 2b.

On the Combiner we write the Keys and Values as we did in the reduce phase of exercise 2b.

On the reduce phase we merge the results we the help of a HashMap on reduce function and later on the cleanup we use this HashMap to write the proper results.