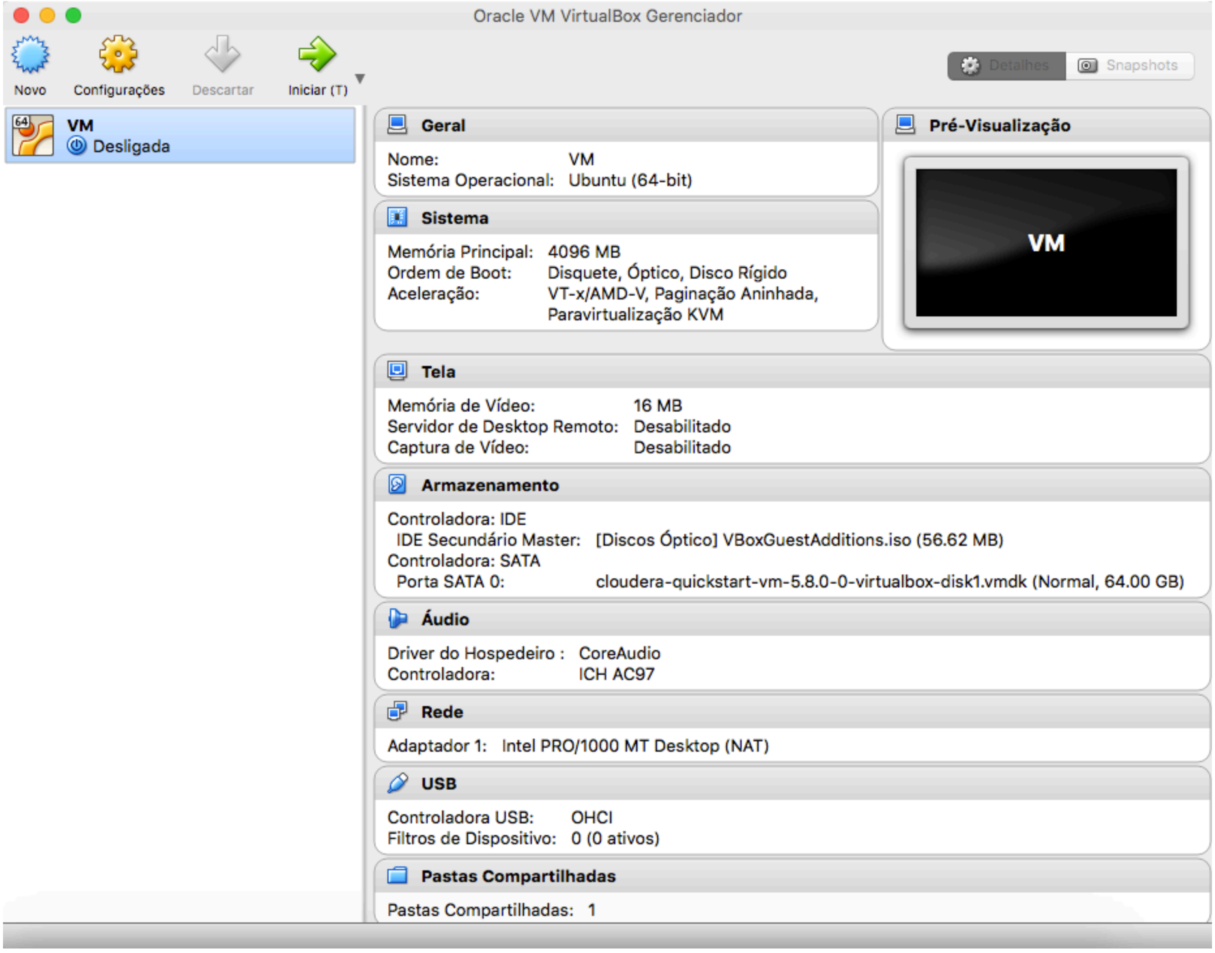
Assignment 1 – Big Data Processing – Olivier Salaün – 16/02/2017.

Settings.

The virtual machine is the same as the one used for the WordCount assignment. It is a CentOS that relies on a 64-bit version of Ubuntu with 4GB of allocated RAM and a 2.5 GHz CPU.

```
[cloudera@quickstart ~]$ hadoop version
Hadoop 2.6.0-cdh5.8.0
Subversion http://github.com/cloudera/hadoop -r
57e7b8556919574d517e874abfb7ebe31a366c2b
Compiled by jenkins on 2016-06-16T19:38Z
Compiled with protoc 2.5.0
From source with checksum 9e99ecd28376acfd5f78c325dd939fed
This command was run using /usr/lib/hadoop/hadoop-common-2.6.0-cdh5.8.0.jar

[cloudera@quickstart ~]$ lsb_release —a
LSB Version:      :base-4.0-amd64:base-4.0-noarch:core-4.0-amd64:core-4.0-noarch
Distributor ID:   CentOS
Description:      CentOS release 6.7 (Final)
Release:    6.7
Codename:   Final

[cloudera@quickstart ~]$ cat /proc/cpuinfo
processor   : 0
vendor_id   : GenuineIntel
cpu family  : 6
model       : 58
model name  : Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz
stepping    : 9
microcode   : 25
cpu MHz           : 2494.316
cache size  : 3072 KB
physical id : 0
siblings    : 1
core id           : 0
cpu cores   : 1
apicid            : 0
initial apicid    : 0
fpu         : yes
fpu_exception     : yes
cpuid level : 13
wp          : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc up rep_good
xtopology nonstop_tsc unfair_spinlock pni pclmulqdq monitor ssse3 cx16 sse4_1
sse4_2 x2apic popcnt aes xsave avx rdrand hypervisor lahf_lm
bogomips    : 4988.63
clflush size      : 64
cache_alignment   : 64
address sizes     : 36 bits physical, 48 bits virtual
power management:
```



Geral

Nome: VM
Sistema Operacional: Ubuntu (64-bit)

Sistema

Memória Principal: 4096 MB
Ordem de Boot: Disquete, Óptico, Disco Rígido
Aceleração: VT-x/AMD-V, Paginação Aninhada, Paravirtualização KVM

Tela

Memória de Vídeo: 16 MB
Servidor de Desktop Remoto: Desabilitado
Captura de Vídeo: Desabilitado

Armazenamento

Controladora: IDE
IDE Secundário Master: [Discos Óptico] VBoxGuestAdditions.iso (56.62 MB)
Controladora: SATA
Porta SATA 0: cloudera-quickstart-vm-5.8.0-0-virtualbox-disk1.vmdk (Normal, 64.00 GB)

Áudio

Driver do Hospedeiro : CoreAudio
Controladora: ICH AC97

Rede

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)

USB

Controladora USB: OHCI
Filtros de Dispositivo: 0 (0 ativos)

Pastas Compartilhadas

Pastas Compartilhadas: 1

In order to keep things clear, we create a whole new folder for each question in Eclipse Package Explorer.

(a) (30) Run a MapReduce program to identify stop words (words with frequency > 4000) for the given document corpus. Store them in a single csv file on HDFS (stopwords.csv). You can edit the several parts of the reducers' output after the job finishes (with hdfs commands or with a text editor), in order to merge them as a single csv file.

For all questions a), the (key, values) formats are the following:
mapper input: (LongWritable, Text) [file names and content of the files]
mapper output – reducer input: (Text, IntWritable) [word, number of occurrences of that word]
reducer output: (Text, IntWritable) [word, number of occurrences of that word]

i. (10) Use 10 reducers and do not use a combiner. Report the execution time.

We create an input folder where all three texts are initially stored and we reuse the code from the WordCount assignment for this question. In the driver class, we add 10 reducers with the following line :

```
job.setNumReduceTasks(10);
```

We also add an if-condition so that existing arguments files are replaced by new ones when the program is executed repeatedly.

```
FileSystem fs = FileSystem.newInstance(getConf());

if (fs.exists(new Path(args[1]))) {
fs.delete(new Path(args[1]), true);
}
```

Since the output is a csv file, we use semicolon as a separator:

```
job.getConfiguration().set("mapreduce.output.textoutputformat.sepa
rator", " ; ");
```

In the map class, we use a for-loop that splits the text at whitespaces between words and convert all words into lowercase strings.

```
for (String string: value.toString().split("\\s+")) {
     word.set(string.toLowerCase());
     context.write(word, ONE);
}
```

In the reduce class, we create another loop that generates a counter for each word and those that have more than 4000 occurrences are included within the output file.

```
for (IntWritable val : values) {
     sum = sum + val.get();
}
if (sum > 4000){
```

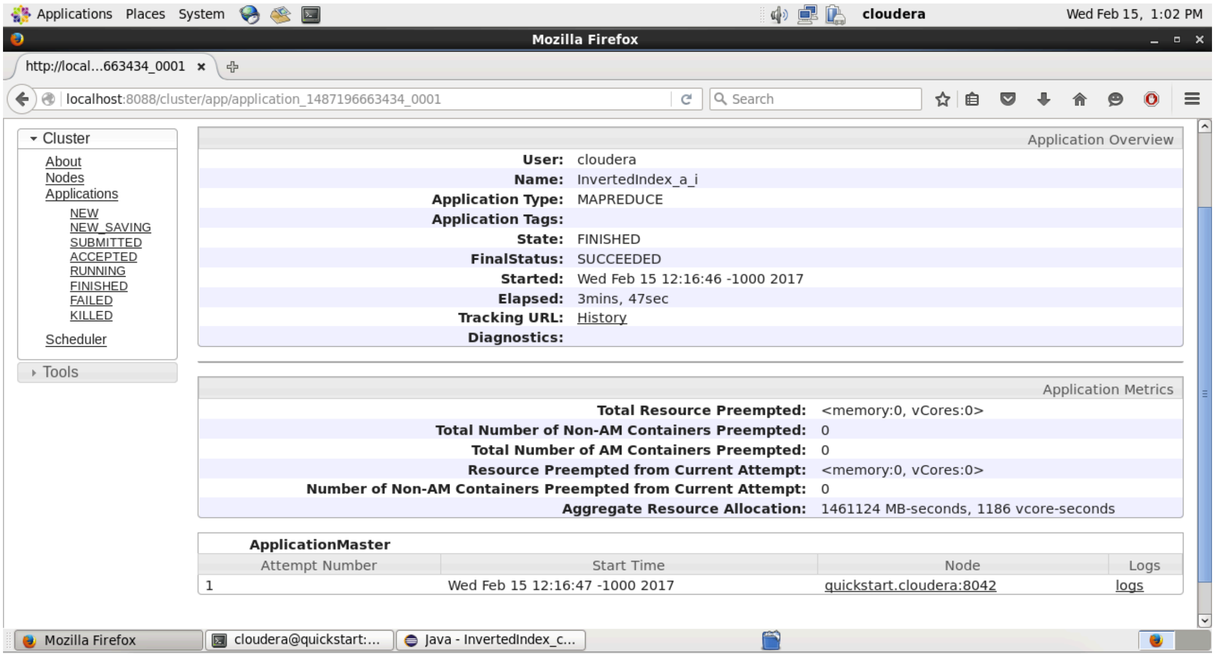
In the outputfile, each key (stop word) comes with its value (integer, number of occurences):

```
context.write(key, new IntWritable(sum));
}
```

We use the following commands in the terminal :

```
cd workspace
hdfs dfsadmin -safemode leave
hadoop jar InvertedIndex_a_i.jar stopwordcount.InvertedIndex_a_i
input output
hadoop fs -ls output
hadoop fs -getmerge output
/home/cloudera/workspace/InvertedIndex_a_i/output/StopWords_10redu
c_nocomb.csv
```

The whole process takes 3minutes and 47seconds.



Application Overview

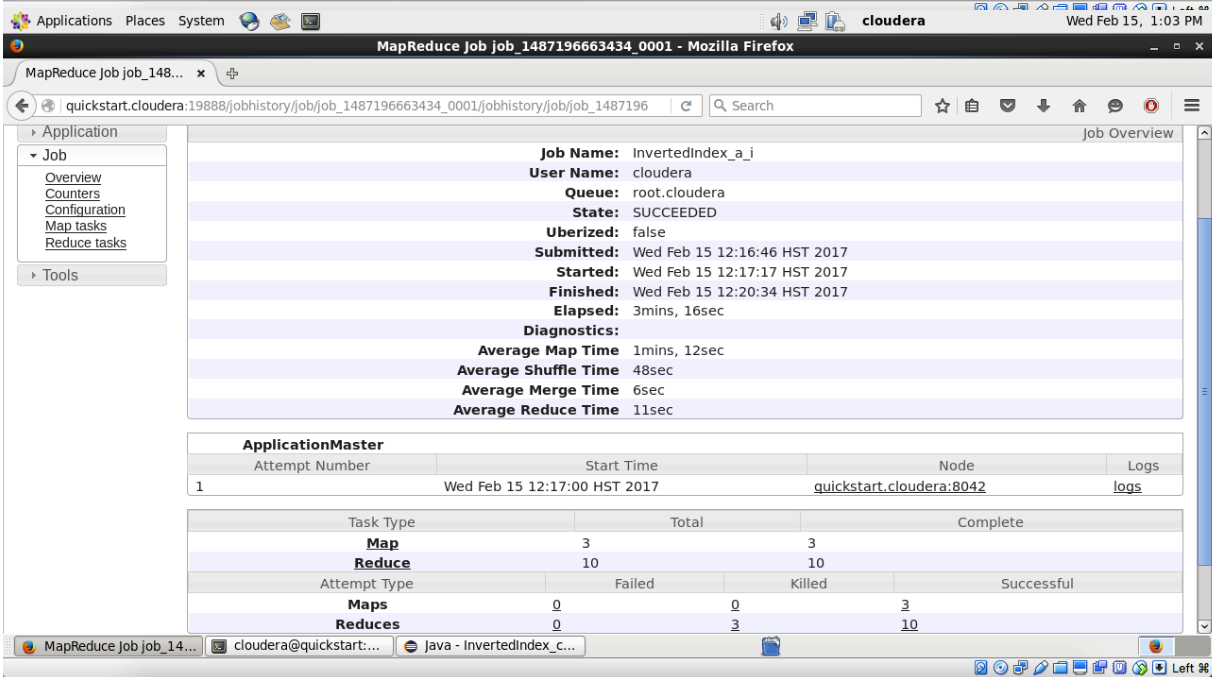
User:	cloudera
Name:	InvertedIndex_a_i
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:16:46 -1000 2017
Elapsed:	3mins, 47sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	1461124 MB-seconds, 1186 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:16:47 -1000 2017	quickstart.cloudera:8042	logs


Applications Places System
cloudera
Wed Feb 15, 1:03 PM
MapReduce Job job_1487196663434_0001 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0001/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name:	InvertedIndex_a_i
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:16:46 HST 2017
Started:	Wed Feb 15 12:17:17 HST 2017
Finished:	Wed Feb 15 12:20:34 HST 2017
Elapsed:	3mins, 16sec
Diagnostics:	
Average Map Time	1mins, 12sec
Average Shuffle Time	48sec
Average Merge Time	6sec
Average Reduce Time	11sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:17:00 HST 2017	quickstart.cloudera:8042	logs

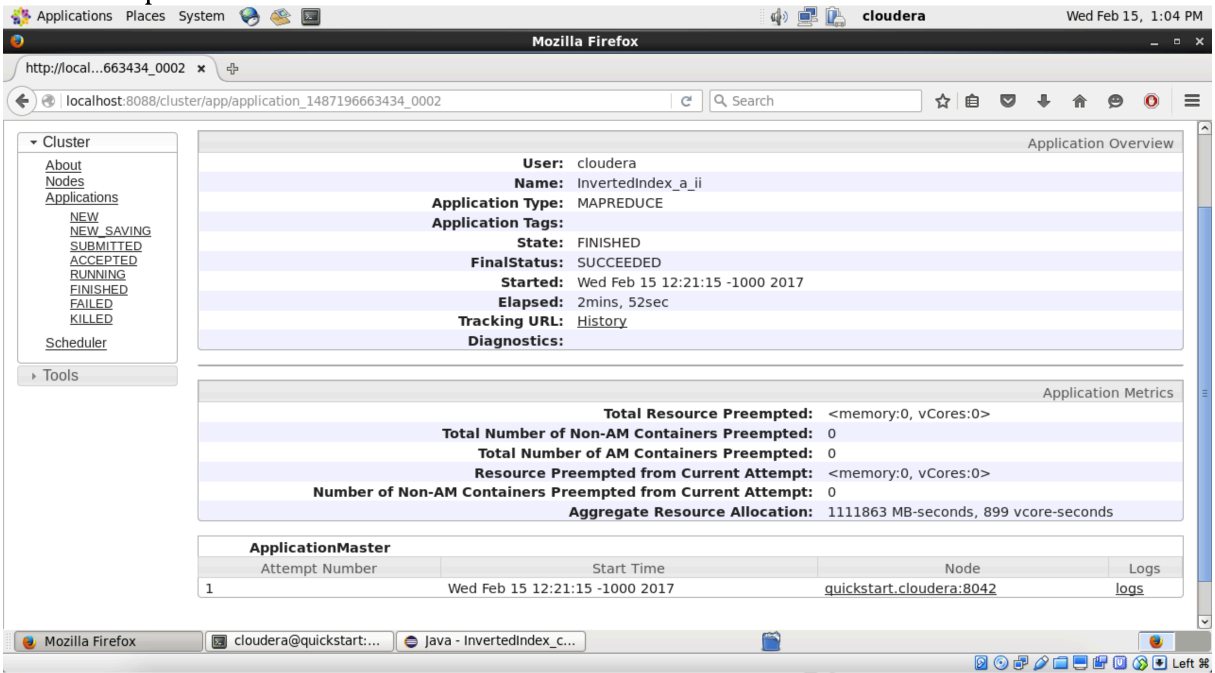
Task Type	Total	Complete	
Map	3	3	
Reduce	10	10	
Attempt Type	Failed	Killed	Successful
Maps	0	0	3
Reduces	0	3	10

ii. (10) Run the same program again, this time using a Combiner. Report the execution time. Is there any difference in the execution time, compared to the previous execution? Why?

The purpose of a combiner is to take the mapper output and to create a reducer input that is minimized. We take the code used previously and add a combiner in the driver class with this line :

```
job.setCombinerClass(Reduce.class);
```

The whole process takes 2 minutes and 52 seconds, showing that the combiner allows a 24% reduction in execution runtime with respect to previous question thanks to minimized reducer input.



Application Overview

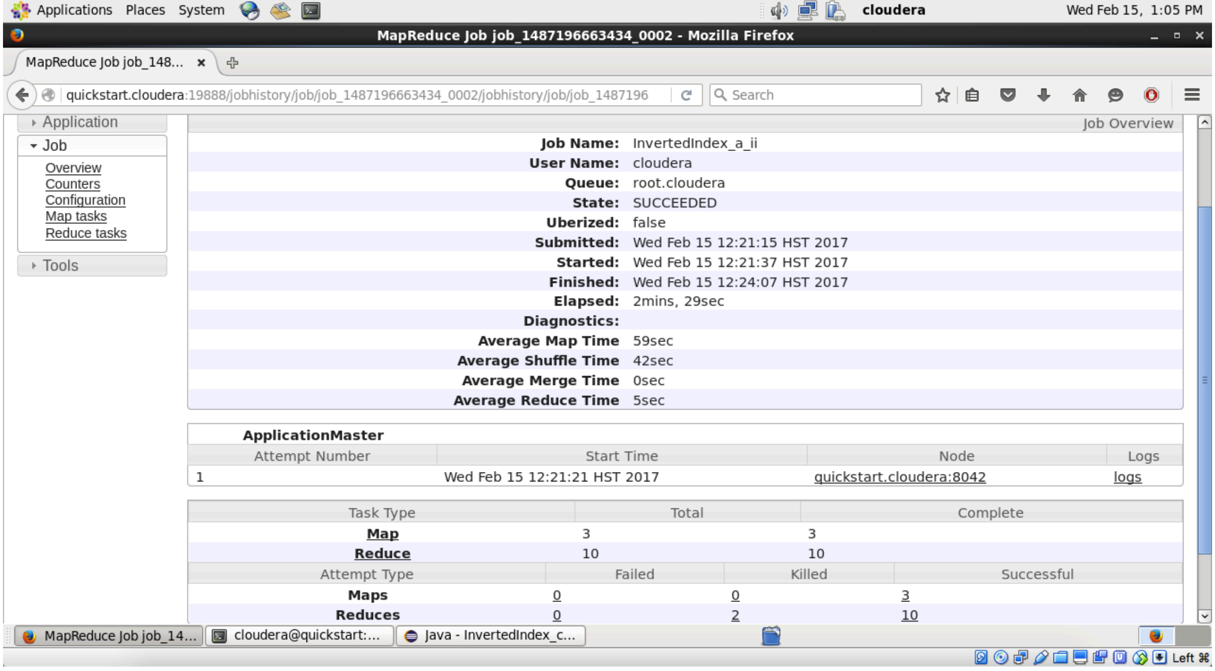
User:	cloudera
Name:	InvertedIndex_a_ii
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:21:15 -1000 2017
Elapsed:	2mins, 52sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	1111863 MB-seconds, 899 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:21:15 -1000 2017	quickstart.cloudera:8042	logs


Applications Places System
cloudera
Wed Feb 15, 1:05 PM
MapReduce Job job_1487196663434_0002 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0002/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name:	InvertedIndex_a_ii
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:21:15 HST 2017
Started:	Wed Feb 15 12:21:37 HST 2017
Finished:	Wed Feb 15 12:24:07 HST 2017
Elapsed:	2mins, 29sec
Diagnostics:	
Average Map Time	59sec
Average Shuffle Time	42sec
Average Merge Time	0sec
Average Reduce Time	5sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:21:21 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	3	3
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	0	3
Reduces	0	2	10

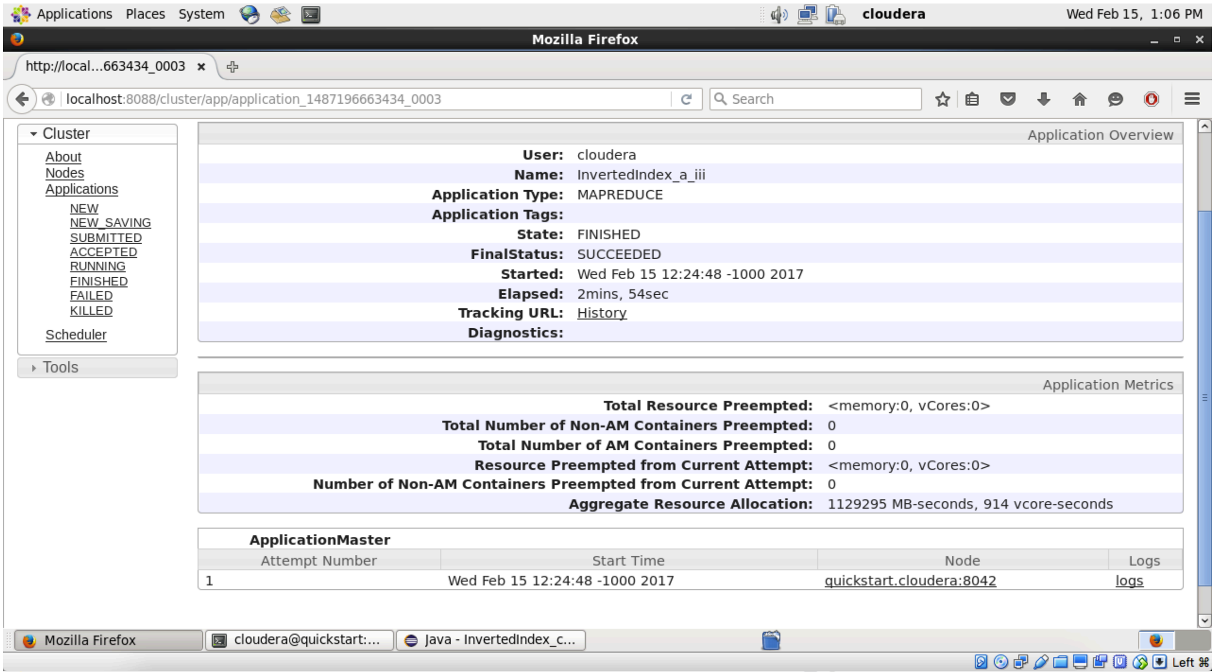
iii. (5) Run the same program again, this time compressing the intermediate results of map (using any codec you wish). Report the execution time. Is there any difference in the execution, time compared to the previous execution? Why?

The purpose of a compressor is to compress input data so that they are faster to read during the hadoop process. It also helps in reducing storage space usage.

We import the following compressors in the code used previously:

```
import org.apache.hadoop.io.compress.BZip2Codec;
import org.apache.hadoop.io.compress.CompressionCodec;
```

The execution time is 2 minutes and 54 seconds, we got almost the same performance like in question a)ii). This might be due to the fact that compression requires significant amount of time. Some codecs do not allow splitting of compressed files but bzip2 supports splitting, so it should not be an issue in our case.



Application Overview

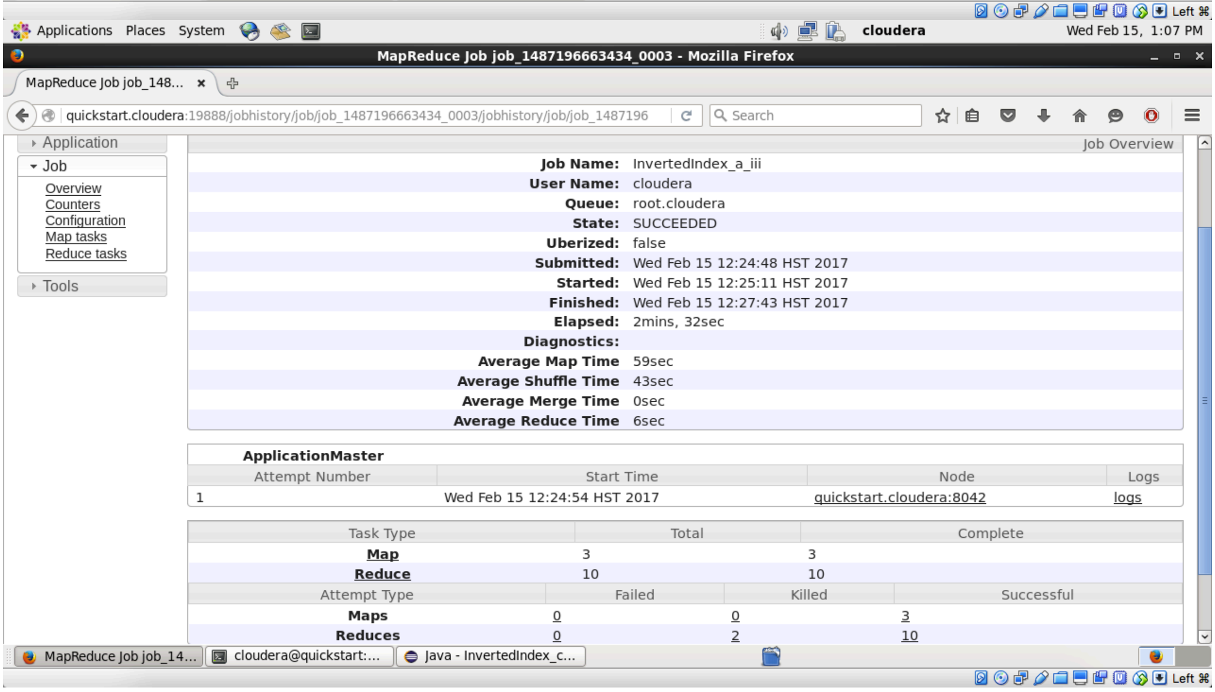
User: cloudera
Name: InvertedIndex_a_iii
Application Type: MAPREDUCE
Application Tags:
State: FINISHED
FinalStatus: SUCCEEDED
Started: Wed Feb 15 12:24:48 -1000 2017
Elapsed: 2mins, 54sec
Tracking URL: History
Diagnostics:

Application Metrics

Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 1129295 MB-seconds, 914 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:24:48 -1000 2017	quickstart.cloudera:8042	logs



Applications Places System
cloudera
Wed Feb 15, 1:07 PM
MapReduce Job job_1487196663434_0003 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0003/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name: InvertedIndex_a_iii
User Name: cloudera
Queue: root.cloudera
State: SUCCEEDED
Uberized: false
Submitted: Wed Feb 15 12:24:48 HST 2017
Started: Wed Feb 15 12:25:11 HST 2017
Finished: Wed Feb 15 12:27:43 HST 2017
Elapsed: 2mins, 32sec
Diagnostics:
Average Map Time 59sec
Average Shuffle Time 43sec
Average Merge Time 0sec
Average Reduce Time 6sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:24:54 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	3	3
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	0	3
Reduces	0	2	10

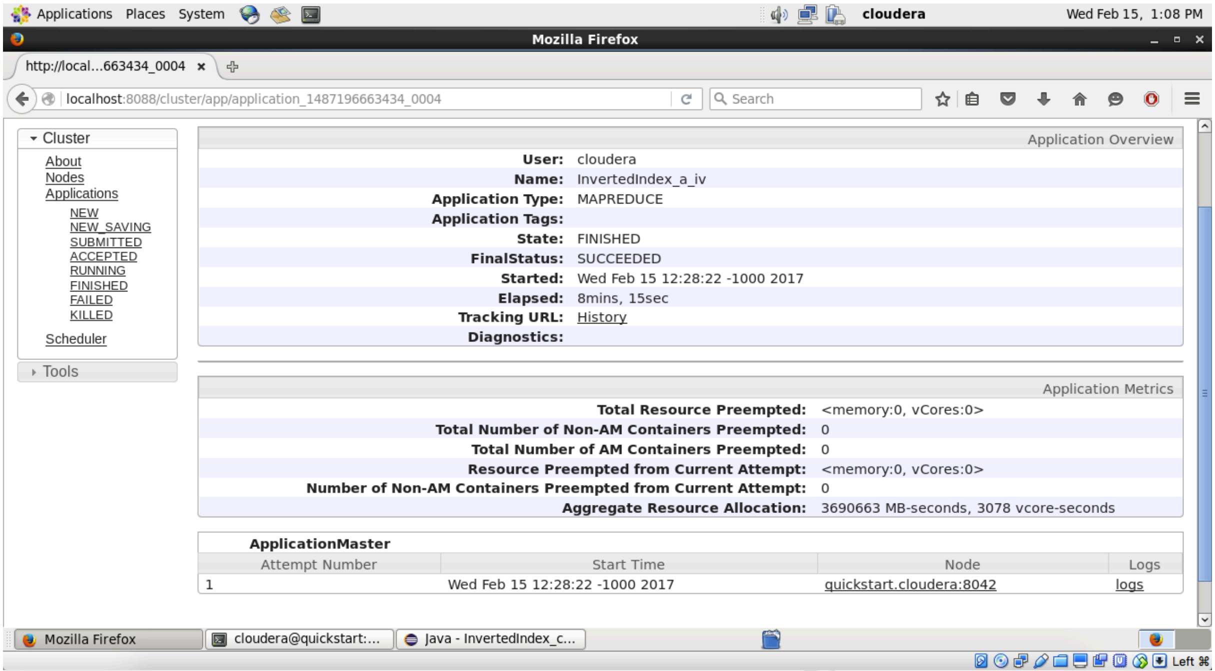
iv. (5) Run the same program again, this time using 50 reducers. Report the execution time. Is there any difference in the execution time, compared to the previous execution? Why?

The purpose of a reducer is to reduce a set of intermediate values corresponding to a common key into a smaller set. Augmenting the number of reducers allows running many reducing tasks in parallel, but the number of reducers has to be a balance:
- if there are too few reducers with respect to the amount of data to process, it implies that more tasks could have been parallelized. The job will be slow with the few existing reducers;
- if the reducers are too many, many tasks will be executed in parallel on very small amount of files, that is time consuming and not optimal.

Unsurprisingly, in this question, setting the number of reducers at 50 causes a dramatic rise in the execution time: 8 minutes and 15 seconds. We used the following line in the driver class:

```
job.setNumReduceTasks(50);
```

We get a 184% increase with respect to the execution time obtained in previous question. There are clearly too many reducers with respect to the amount of files to process.



Application Overview

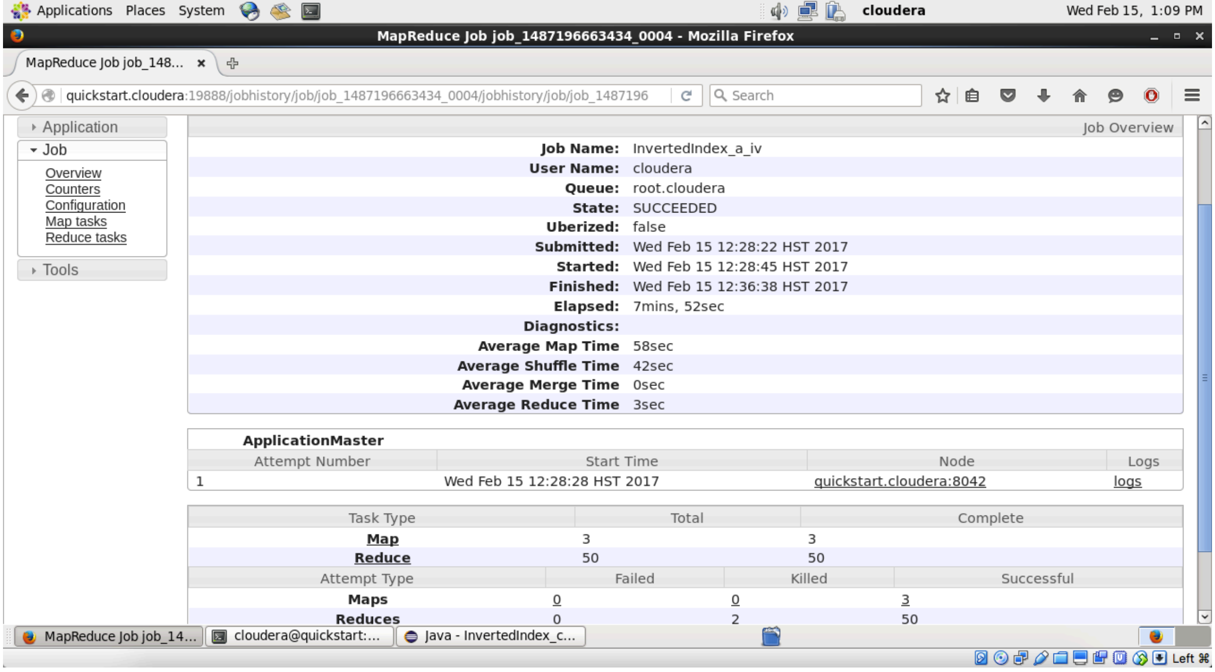
User:	cloudera
Name:	InvertedIndex_a_iv
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:28:22 -1000 2017
Elapsed:	8mins, 15sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	3690663 MB-seconds, 3078 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:28:22 -1000 2017	quickstart.cloudera:8042	logs



Applications Places System
cloudera
Wed Feb 15, 1:09 PM
MapReduce Job job_1487196663434_0004 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0004/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name:	InvertedIndex_a_iv
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:28:22 HST 2017
Started:	Wed Feb 15 12:28:45 HST 2017
Finished:	Wed Feb 15 12:36:38 HST 2017
Elapsed:	7mins, 52sec
Diagnostics:	
Average Map Time	58sec
Average Shuffle Time	42sec
Average Merge Time	0sec
Average Reduce Time	3sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:28:28 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	3	3
Reduce	50	50

Attempt Type	Failed	Killed	Successful
Maps	0	0	3
Reduces	0	2	50

(b) (30) Implement a simple inverted index for the given document corpus, as shown in the previous Table, skipping the words of stopwords.csv.

The (key, values) formats are the following:
mapper input: (LongWritable, Text) [file names and content of the files]
mapper output – reducer input: (Text, Text) [word, file name where the word appears]
reducer output: (Text, Text) [word, file name where the word appears]

We reuse code from previous questions and we set the number of reducers at 10 in the driver class.

```
job.setNumReduceTasks(10);
```

Our folder contains the three text input files and a StopWords.csv obtained from questions a). In the mapper class, we create a HashSet for storing the stop words from the csv file.

```
HashSet<String> stopwordslist = new HashSet<String>();
     BufferedReader Reader = new BufferedReader(
          new FileReader(new
     File("/home/cloudera/workspace/InvertedIndex_b/StopWords.csv"
     )));

String pattern;
while ((pattern = Reader.readLine()) != null) {
     stopwordslist.add(pattern.toLowerCase());
```

We use a for loop in order to exclude the words from the input files that match those contained in the stopwordslist

```
for (String string : value.toString().split("\\s+")) {
     if (!stopwordslist.contains(string.toLowerCase())) {
          word.set(string.toLowerCase());
     }
}
```

The intermediate mapper output files include series of keys (word) with corresponding file(s) name where it is used. The reducer will gather all values corresponding respectively to each key.

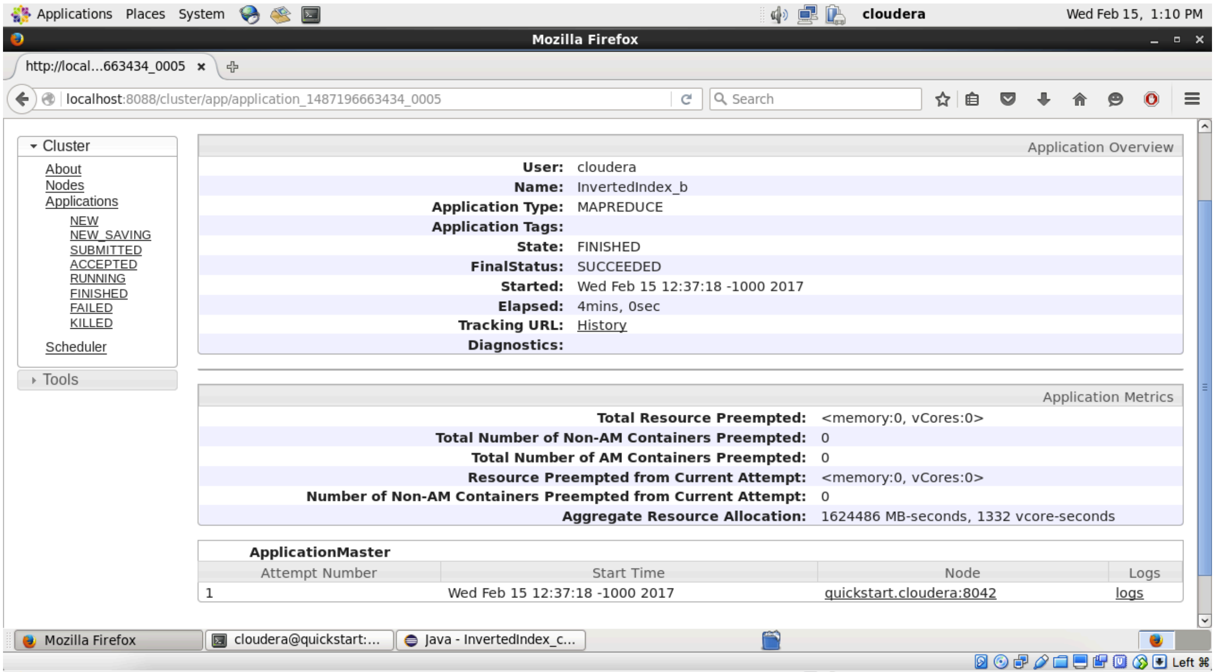
```
context.write(word, filename);
```

In the reducer class, for each key, we add all matching values (files names in which the word is contained). All files names are separated by commas and spaces:

```
HashSet<String> set = new HashSet<String>();

for (Text value : values) {
     set.add(value.toString());
}
```

```
StringBuilder builder = new StringBuilder();

String separator = "";
for (String value : set) {
     builder.append(separator);
     separator = ", ";
     builder.append(value);
}

context.write(key, new Text(builder.toString()));
```

Below are screenshots about the execution runtime:



localhost:8088/cluster/app/application_1487196663434_0005

Application Overview

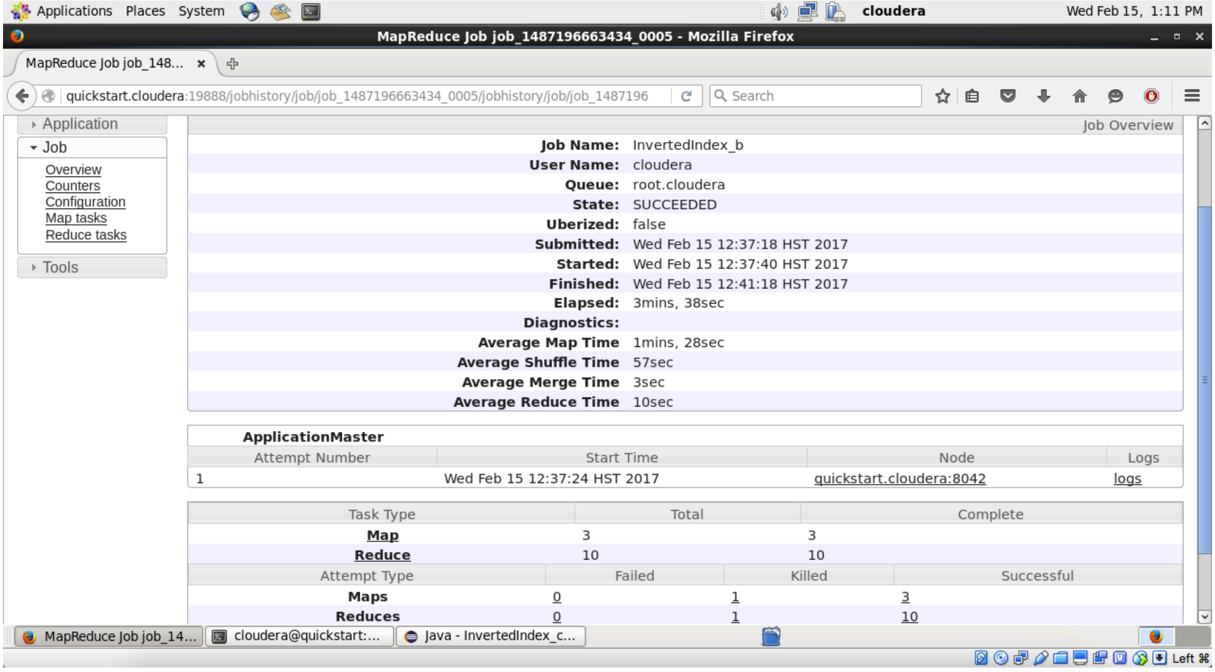
User:	cloudera
Name:	InvertedIndex_b
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:37:18 -1000 2017
Elapsed:	4mins, 0sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	1624486 MB-seconds, 1332 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:37:18 -1000 2017	quickstart.cloudera:8042	logs



quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0005/jobhistory/job/job_1487196

Job Overview

Job Name:	InvertedIndex_b
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:37:18 HST 2017
Started:	Wed Feb 15 12:37:40 HST 2017
Finished:	Wed Feb 15 12:41:18 HST 2017
Elapsed:	3mins, 38sec
Diagnostics:	
Average Map Time	1mins, 28sec
Average Shuffle Time	57sec
Average Merge Time	3sec
Average Reduce Time	10sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:37:24 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete	
Map	3	3	
Reduce	10	10	
Attempt Type	Failed	Killed	Successful
Maps	0	1	3
Reduces	0	1	10

(c) (10) How many unique words exist in the document corpus (excluding stop words)? Which counter(s) reveal(s) this information? Define your own counter for the number of words appearing in a single document only. What is the value of this counter? Store the final value of this counter on a new file on HDFS.

In the code used in question b), we add the following line just before the driver class:

```
public static enum UNIQUE_WORDS_COUNTER {
     NUMBER_OF_UNIQUE_WORDS,
};
```

In the mapper class, stop words are excluded from the mapper output. In the reducer class, each word is added to the HashSet set. The UNIQUE_WORD_COUNTER increases by one for each unique word that appears in the document, with the following if-condition:

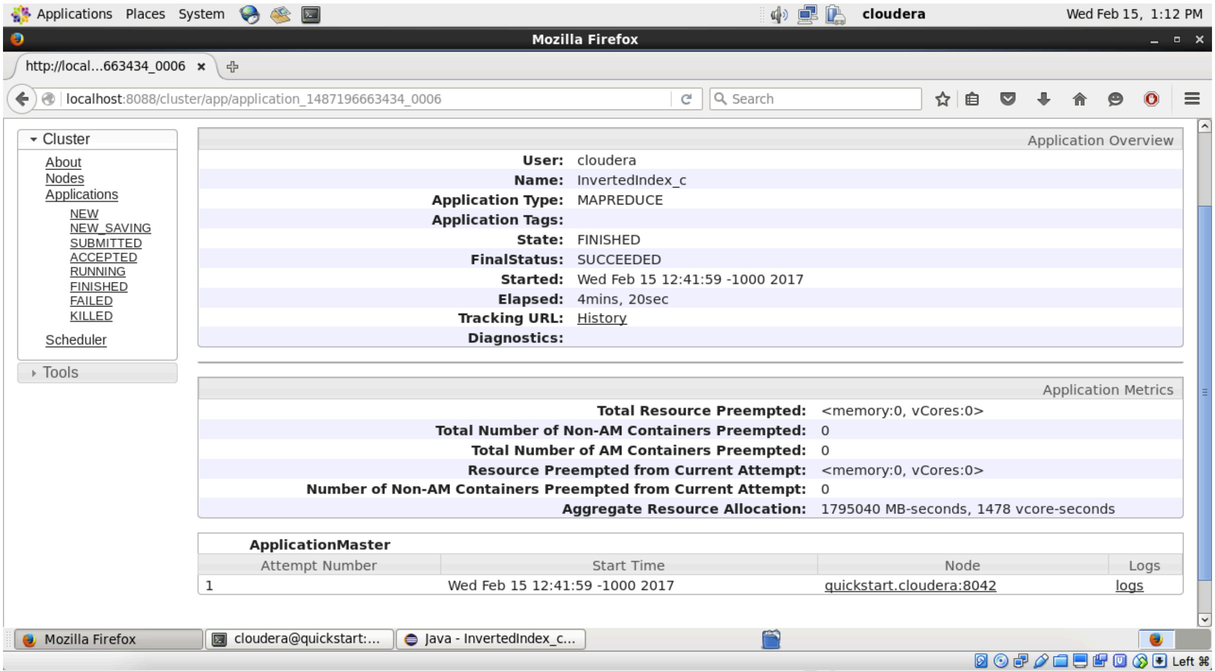
```
if (set.size() == 1) {
     context.getCounter(UNIQUE_WORDS_COUNTER.NUMBER_OF_UNIQUE_WORD
S).increment(1);
```

The total number of unique words appears at the end of the terminal output. We end up with 57033 unique words:

```
[cloudera@quickstart workspace]$ hadoop jar InvertedIndex_c.jar
stopwordcount.InvertedIndex_c input output
[input, output]
[input, output]
17/02/15 12:41:53 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/02/15 12:41:57 INFO input.FileInputFormat: Total input paths to process : 3
17/02/15 12:41:57 INFO mapreduce.JobSubmitter: number of splits:3
17/02/15 12:41:58 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1487196663434_0006
17/02/15 12:41:59 INFO impl.YarnClientImpl: Submitted application
application_1487196663434_0006
17/02/15 12:41:59 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1487196663434_0006/
17/02/15 12:41:59 INFO mapreduce.Job: Running job: job_1487196663434_0006
17/02/15 12:42:22 INFO mapreduce.Job: Job job_1487196663434_0006 running in uber mode :
false
17/02/15 12:42:22 INFO mapreduce.Job:  map 0% reduce 0%
17/02/15 12:42:55 INFO mapreduce.Job:  map 1% reduce 0%
17/02/15 12:42:59 INFO mapreduce.Job:  map 3% reduce 0%
17/02/15 12:43:01 INFO mapreduce.Job:  map 4% reduce 0%
[…]
17/02/15 12:46:07 INFO mapreduce.Job:  map 100% reduce 87%
17/02/15 12:46:10 INFO mapreduce.Job:  map 100% reduce 88%
17/02/15 12:46:14 INFO mapreduce.Job:  map 100% reduce 93%
17/02/15 12:46:18 INFO mapreduce.Job:  map 100% reduce 100%
17/02/15 12:46:21 INFO mapreduce.Job: Job job_1487196663434_0006 completed successfully
17/02/15 12:46:22 INFO mapreduce.Job: Counters: 52
	File System Counters
		FILE: Number of bytes read=9334569
		FILE: Number of bytes written=20190356
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=25755981
		HDFS: Number of bytes written=1263324
		HDFS: Number of read operations=39
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=20
	Job Counters
```

```
                Killed map tasks=1
                Killed reduce tasks=2
                Launched map tasks=4
                Launched reduce tasks=11
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=350410
                Total time spent by all reduces in occupied slots (ms)=841723
                Total time spent by all map tasks (ms)=350410
                Total time spent by all reduce tasks (ms)=841723
                Total vcore-seconds taken by all map tasks=350410
                Total vcore-seconds taken by all reduce tasks=841723
                Total megabyte-seconds taken by all map tasks=358819840
                Total megabyte-seconds taken by all reduce tasks=861924352
        Map-Reduce Framework
                Map input records=507536
                Map output records=507536
                Map output bytes=8319437
                Map output materialized bytes=9334689
                Input split bytes=381
                Combine input records=0
                Combine output records=0
                Reduce input groups=71037
                Reduce shuffle bytes=9334689
                Reduce input records=507536
                Reduce output records=57033
                Spilled Records=1015072
                Shuffled Maps =30
                Failed Shuffles=0
                Merged Map outputs=30
                GC time elapsed (ms)=5425
                CPU time spent (ms)=58300
                Physical memory (bytes) snapshot=1796476928
                Virtual memory (bytes) snapshot=19568939008
                Total committed heap usage (bytes)=1104359424
        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=25755600
        File Output Format Counters
                Bytes Written=1263324
        stopwordcount.InvertedIndex_c$UNIQUE_WORDS_COUNTER
                NUMBER_OF_UNIQUE_WORDS=57033
```

Below are screenshots for the execution runtime.



Application Overview

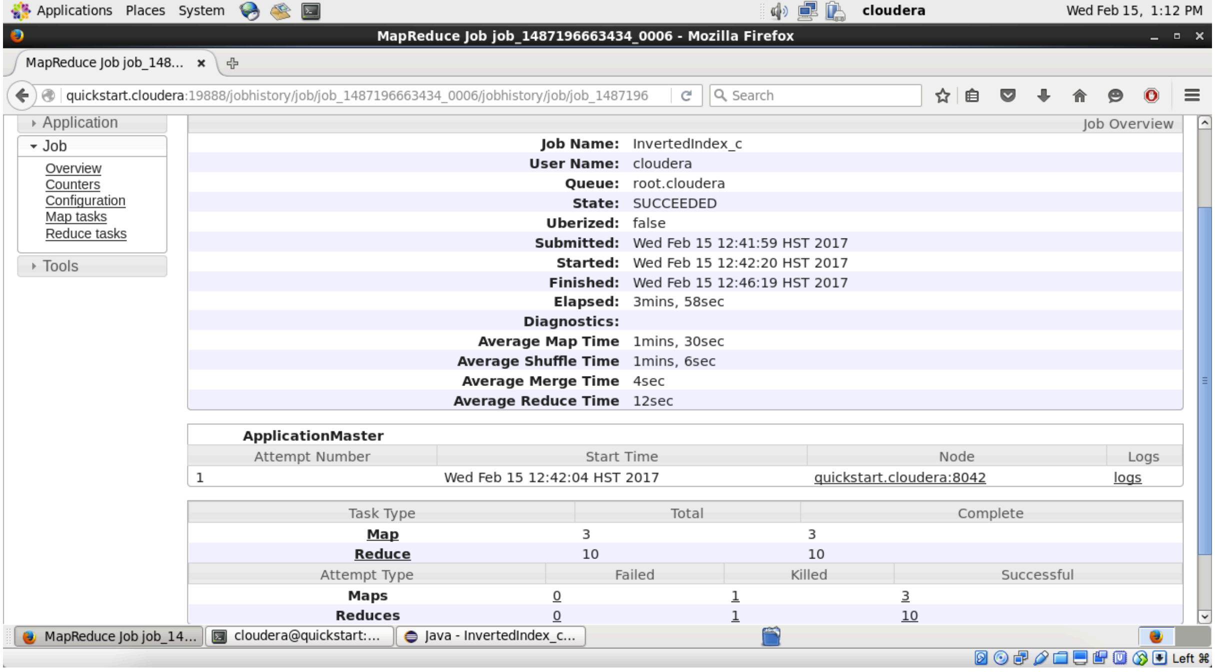
User:	cloudera
Name:	InvertedIndex_c
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:41:59 -1000 2017
Elapsed:	4mins, 20sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	1795040 MB-seconds, 1478 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:41:59 -1000 2017	quickstart.cloudera:8042	logs


Applications Places System
cloudera
Wed Feb 15, 1:12 PM
MapReduce Job job_1487196663434_0006 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0006/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name:	InvertedIndex_c
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:41:59 HST 2017
Started:	Wed Feb 15 12:42:20 HST 2017
Finished:	Wed Feb 15 12:46:19 HST 2017
Elapsed:	3mins, 58sec
Diagnostics:	
Average Map Time	1mins, 30sec
Average Shuffle Time	1mins, 6sec
Average Merge Time	4sec
Average Reduce Time	12sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:42:04 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete	
Map	3	3	
Reduce	10	10	
Attempt Type	Failed	Killed	Successful
Maps	0	1	3
Reduces	0	1	10

(d) (30) Extend the inverted index of (b), in order to keep the frequency of each word for each document. The new output should be of the form: [table on questions sheet] which means that the word frequency should follow a single '#' character, which should follow the filename, for each file that contains this word. You are required to use a Combiner.

In this question, the (key, values) formats are the following:
mapper input: (LongWritable, Text) [file names, text of the files]
mapper output – combiner input: (Text, Text) [word, file(s) name(s) where the word appears]
mapper output – combiner input: (Text, Text) [word, file(s) name(s) where the word appears]
combiner output – reducer input: (Text, Text) [word, file(s) name(s) where the word appears]
reducer output: (Text, Text) [word, file(s) name(s) where the word appears]

We reuse the code made previously and add a Combiner class. In the Reducer class, we use a HashMap that can contain a string with an integer.

```
HashMap<String, Integer> container = new HashMap<String,
Integer>();
```

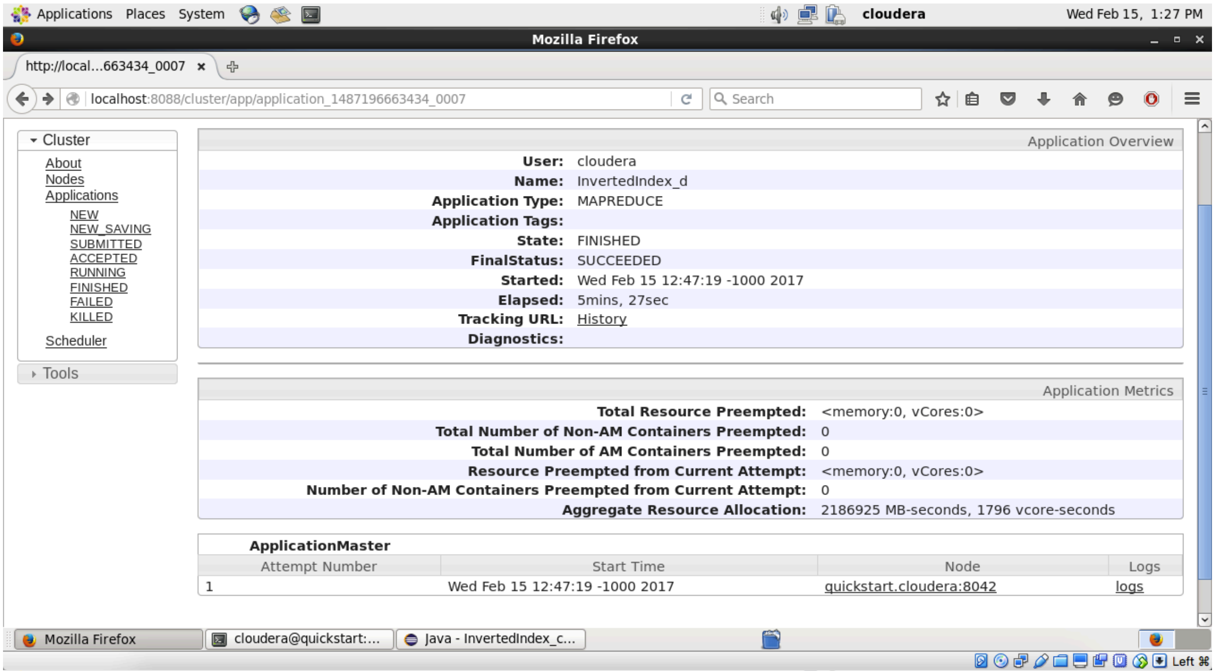
For each key/word, it will store files names (string) with the corresponding number of occurrences (integer). With a for-loop and an if-else condition, for each key, we span the values (file names), store them, if not already present, in the HashMap and set the corresponding integer value at 1. In case the file name was already stored, we increase the corresponding integer of the HashMap by 1.

```
for (Text value : values) {
if (!container.containsKey(value.toString())) {
      container.put(value.toString(),1);

}else{

      int count = container.get(value.toString()) + 1;
      container.put(value.toString(), count);
}
}
```

Finally, when generating the output file, for each key (word), we append the file(s) name(s) where the word appears and the number of occurrences that was stored in the Hashmap.

```
StringBuilder builder = new StringBuilder();

String separator = "";
for (String value : container.keySet()) {
     builder.append(separator);
     separator = "; ";
     builder.append(value + "#" + container.get(value));
}

context.write(key, new Text(builder.toString()));
```

Below are the screenshots with the execution time:



Application Overview

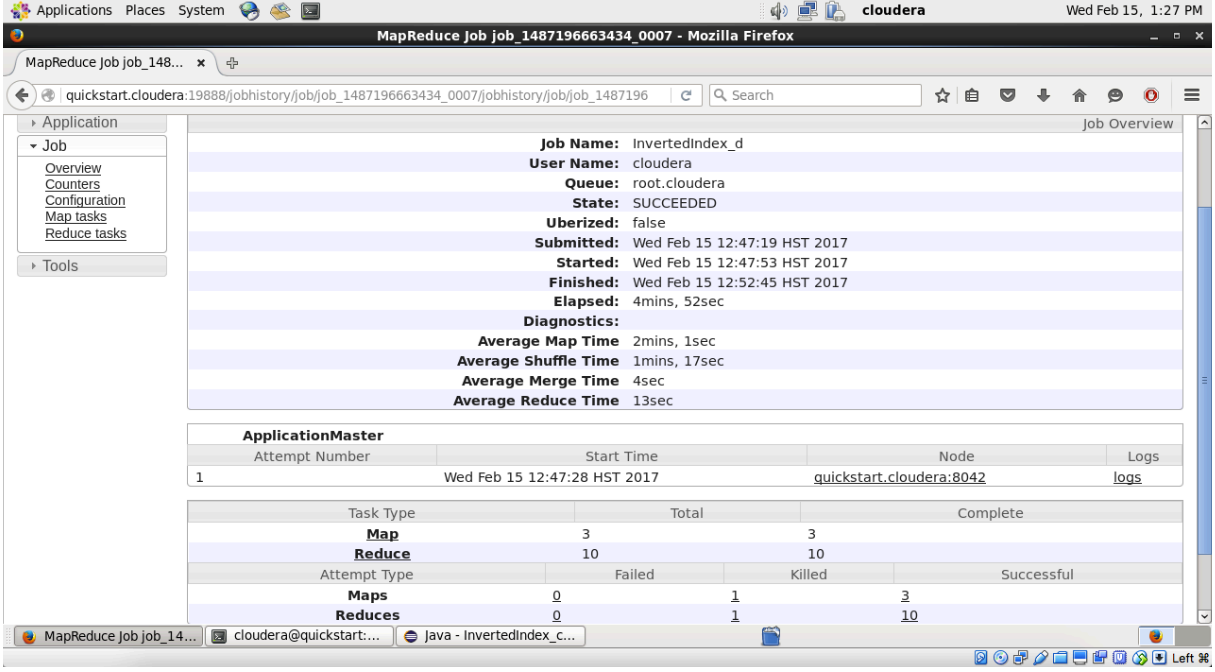
User:	cloudera
Name:	InvertedIndex_d
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Wed Feb 15 12:47:19 -1000 2017
Elapsed:	5mins, 27sec
Tracking URL:	History
Diagnostics:	

Application Metrics

Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	2186925 MB-seconds, 1796 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:47:19 -1000 2017	quickstart.cloudera:8042	logs


Applications Places System
cloudera
Wed Feb 15, 1:27 PM
MapReduce Job job_1487196663434_0007 - Mozilla Firefox
MapReduce Job job_148...
quickstart.cloudera:19888/jobhistory/job/job_1487196663434_0007/jobhistory/job/job_1487196
Search
Application
Job
Overview
Counters
Configuration
Map tasks
Reduce tasks
Tools
MapReduce Job job_14...
cloudera@quickstart:...
Java - InvertedIndex_c...
Left ⌘

Job Overview

Job Name:	InvertedIndex_d
User Name:	cloudera
Queue:	root.cloudera
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 15 12:47:19 HST 2017
Started:	Wed Feb 15 12:47:53 HST 2017
Finished:	Wed Feb 15 12:52:45 HST 2017
Elapsed:	4mins, 52sec
Diagnostics:	
Average Map Time	2mins, 1sec
Average Shuffle Time	1mins, 17sec
Average Merge Time	4sec
Average Reduce Time	13sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 15 12:47:28 HST 2017	quickstart.cloudera:8042	logs

Task Type	Total	Complete
Map	3	3
Reduce	10	10

Attempt Type	Failed	Killed	Successful
Maps	0	1	3
Reduces	0	1	10

Bibliography.

White, Tom E. *Hadoop: The Definitive Guide (4th Edition)*. N.p.: O'Reilly Media, 2015.