

# Massive Data Processing - Lab 1

Emilie Leblanc, emilie.leblanc@student.ecp.fr

February 21, 2017

## 1 Hadoop setup

For my Hadoop setup, I have used a VirtualBox in which I have installed Ubuntu 16.04 and Java. I have also created a new user, **hduser**, on which I installed Hadoop 2.7.3.

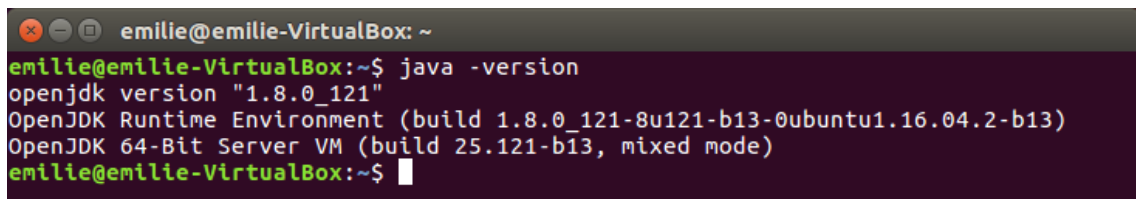
A terminal window titled 'emilie@emilie-VirtualBox: ~' with a dark background. The prompt is 'emilie@emilie-VirtualBox:~\$'. The command 'java -version' has been executed, resulting in the following output: 'openjdk version "1.8.0\_121"', 'OpenJDK Runtime Environment (build 1.8.0\_121-8u121-b13-0ubuntu1.16.04.2-b13)', and 'OpenJDK 64-Bit Server VM (build 25.121-b13, mixed mode)'. The prompt is now 'emilie@emilie-VirtualBox:~\$' with a cursor.

Figure 1: Version of Java installed.

In order to set up Hadoop, I followed various tutorials, including (but not restricted to):

- <https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-16-04>
- <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>
- [https://www.youtube.com/watch?v=ve841JxF\\_VE](https://www.youtube.com/watch?v=ve841JxF_VE)

In these tutorials, I understood how to setup the different Hadoop files. In **hadoop-env.sh**, I set:

```
1 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/
```

In **etc/hadoop/core-site.xml**, I set:

```
1 <configuration>
2   <property>
3     <name>fs.defaultFS</name>
4     <value>hdfs://localhost:9000</value>
5   </property>
6 </configuration>
```

In **etc/hadoop/hdfs-site.xml**, I set:

```
1 <configuration>
2   <property>
3     <name>dfs.replication</name>
4     <value>1</value>
5   </property>
6 </configuration>
```

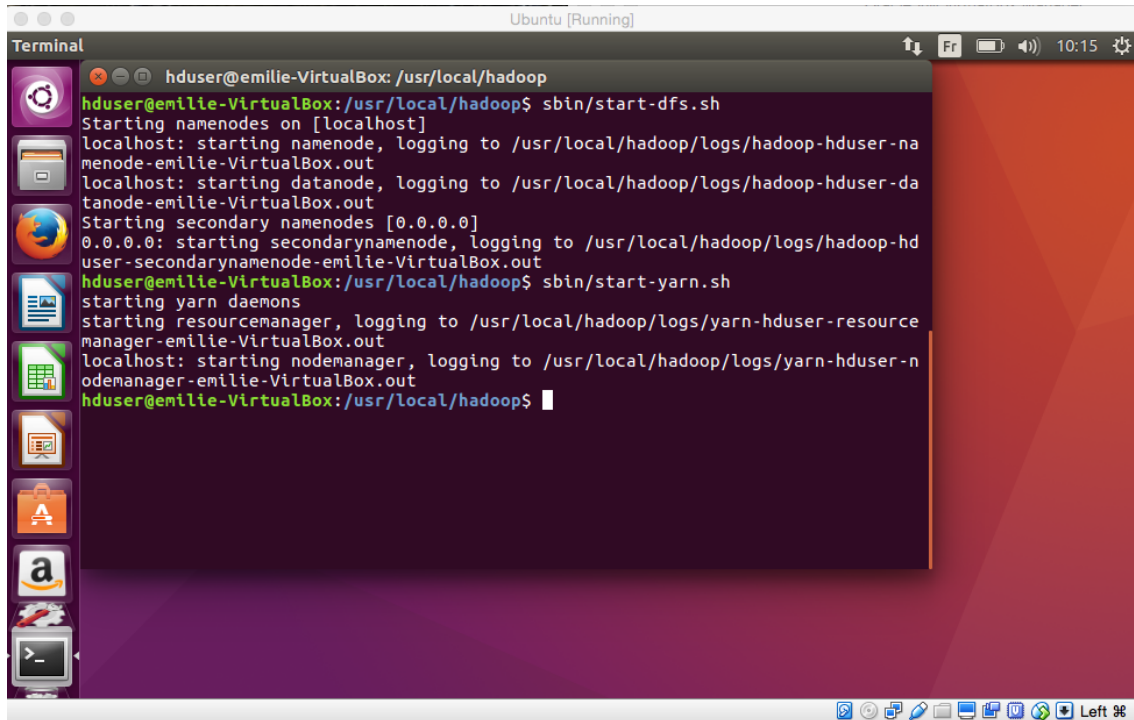
I also create the **etc/hadoop/mapred-site.xml** file as:

```
1 <configuration>
2   <property>
3     <name>mapreduce.framework.name</name>
4     <value>yarn</value>
5   </property>
6 </configuration>
```

And configured `etc/hadoop/yarn-site.xml` to:

```
1 <configuration>
2   <property>
3     <name>yarn.nodemanager.aux-services</name>
4     <value>mapreduce_shuffle</value>
5   </property>
6 </configuration>
```

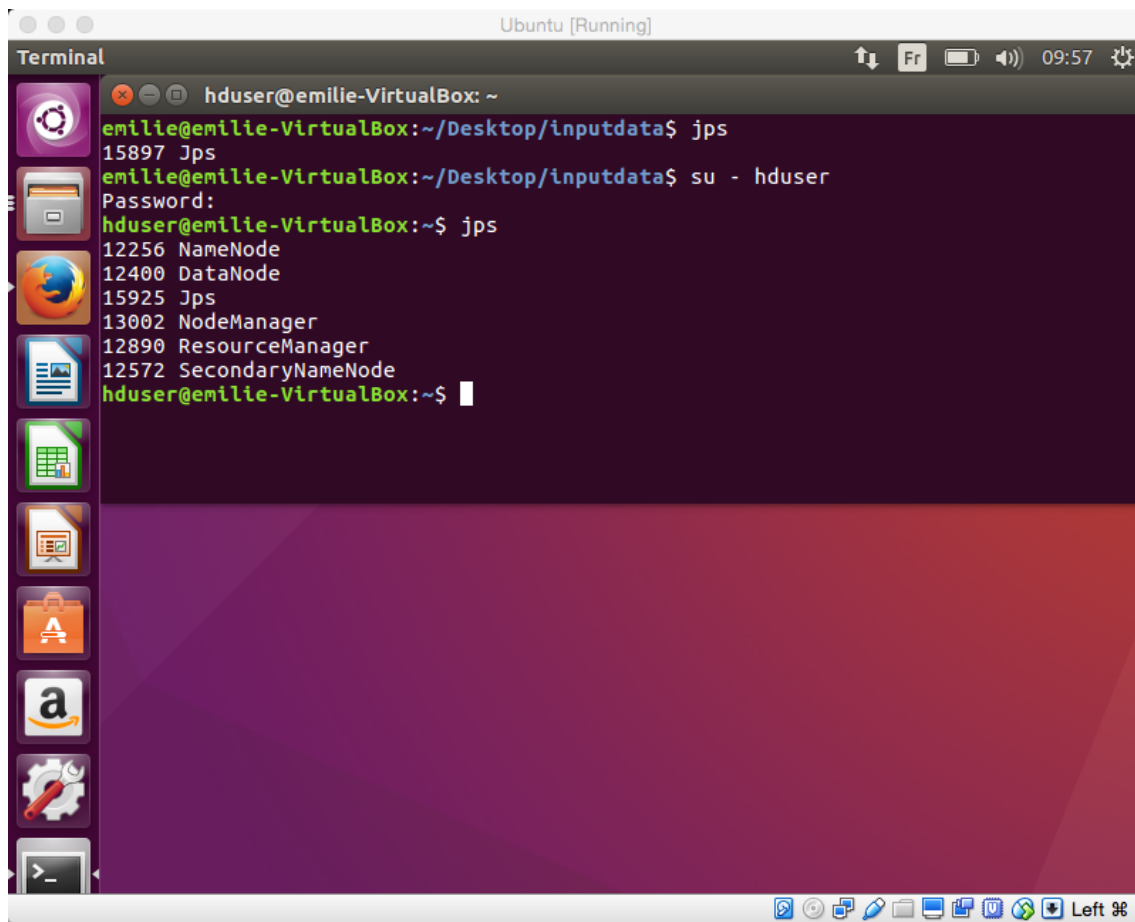
In the `hduser`, after having formatted the filesystem with the `bin/hdfs namenode -format` command line, I started the `start-dfs.sh` and the `start-yarn.sh` files as seen below.

A screenshot of a terminal window titled "Terminal" with a dark background. The prompt is "hduser@emilie-VirtualBox: /usr/local/hadoop". The user has executed "sbin/start-dfs.sh", which outputs: "Starting namenodes on [localhost]", "localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-emilie-VirtualBox.out", "localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-emilie-VirtualBox.out", and "Starting secondary namenodes [0.0.0.0]". Then, the user executed "sbin/start-yarn.sh", which outputs: "starting yarn daemons", "starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-emilie-VirtualBox.out", and "localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-emilie-VirtualBox.out". The terminal window is part of an Ubuntu desktop environment, with a sidebar on the left showing various application icons and a system tray at the bottom right showing the time as 10:15.

```
hduser@emilie-VirtualBox: /usr/local/hadoop
hduser@emilie-VirtualBox: /usr/local/hadoop$ sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-na
menode-emilie-VirtualBox.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-da
tanode-emilie-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-emilie-VirtualBox.out
hduser@emilie-VirtualBox: /usr/local/hadoop$ sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-emilie-VirtualBox.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-emilie-VirtualBox.out
hduser@emilie-VirtualBox: /usr/local/hadoop$
```

Figure 2: Starting processes.

And then verified that everything was running correctly with the `jps` command:



The image shows a terminal window titled "Terminal" within an "Ubuntu [Running]" virtual machine. The prompt is "hduser@emilie-VirtualBox: ~". The user enters "emilie@emilie-VirtualBox:~/Desktop/inputdata\$ jps", which returns "15897 Jps". Then, the user enters "emilie@emilie-VirtualBox:~/Desktop/inputdata\$ su - hduser", prompts for a password, and then enters "hduser@emilie-VirtualBox:~\$ jps". This returns a list of running Hadoop processes: "12256 NameNode", "12400 DataNode", "15925 Jps", "13002 NodeManager", "12890 ResourceManager", and "12572 SecondaryNameNode". The prompt returns to "hduser@emilie-VirtualBox:~\$". The terminal window has a dark purple background and a sidebar with application icons on the left. The system tray at the bottom shows various icons and the text "Left %".

```
hduser@emilie-VirtualBox: ~
emilie@emilie-VirtualBox:~/Desktop/inputdata$ jps
15897 Jps
emilie@emilie-VirtualBox:~/Desktop/inputdata$ su - hduser
Password:
hduser@emilie-VirtualBox:~$ jps
12256 NameNode
12400 DataNode
15925 Jps
13002 NodeManager
12890 ResourceManager
12572 SecondaryNameNode
hduser@emilie-VirtualBox:~$
```

Figure 3: Jps command.

After having configured Hadoop correctly, I downloaded the datasets.

```
emilie@emilie-VirtualBox: ~/Desktop/inputdata
emilie@emilie-VirtualBox:~/Desktop/inputdata$ curl http://www.gutenberg.org/cache/epub/100/pg100.txt > pg100.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 5458k  100 5458k    0     0  953k      0  0:00:05  0:00:05 --:--:-- 1116k
emilie@emilie-VirtualBox:~/Desktop/inputdata$ curl http://www.gutenberg.org/cache/epub/31100/pg31100.txt > pg31100.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 4349k  100 4349k    0     0  831k      0  0:00:05  0:00:05 --:--:--  915k
emilie@emilie-VirtualBox:~/Desktop/inputdata$ curl http://www.gutenberg.org/cache/epub/3200/pg3200.txt > pg3200.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 15.2M  100 15.2M    0     0 1407k      0  0:00:11  0:00:11 --:--:-- 1660k
emilie@emilie-VirtualBox:~/Desktop/inputdata$
```

Figure 4: Downloading datasets from gutenber.

## 2 Run a MapReduce to identify stopwords

In order to run a MapReduce to identify stopwords, I created a **all.txt** file, containing all the lines of the **pg100.txt**, **pg31100.txt** and **pg3200.txt** files through the following command line:

```
1 cat input-files/* > all.txt
```

Then, I created a user directory in the HDFS and inserted the data.

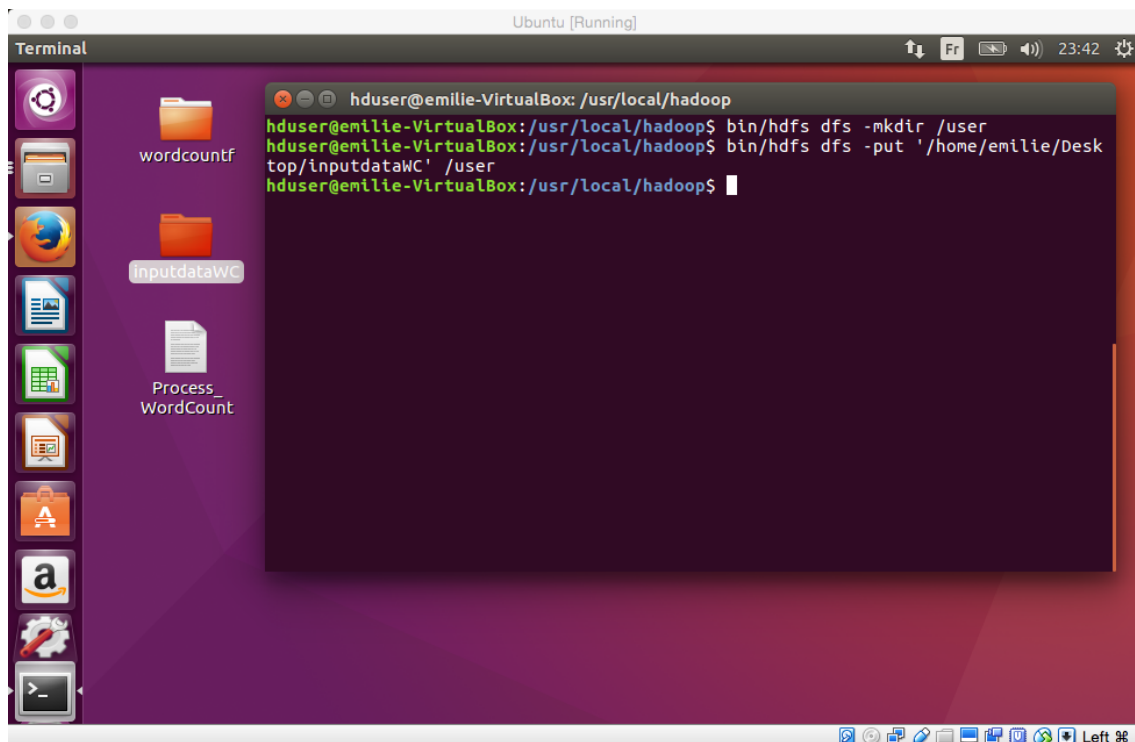


Figure 5: Creating directory in HDFS and inserting data.

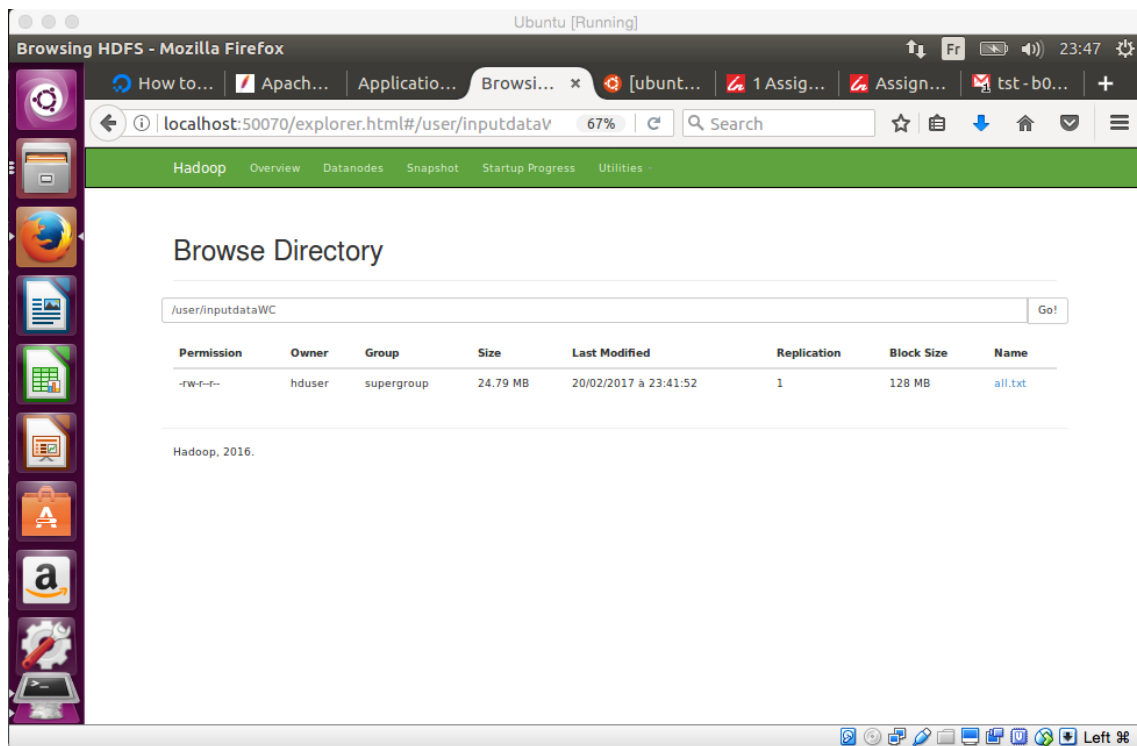
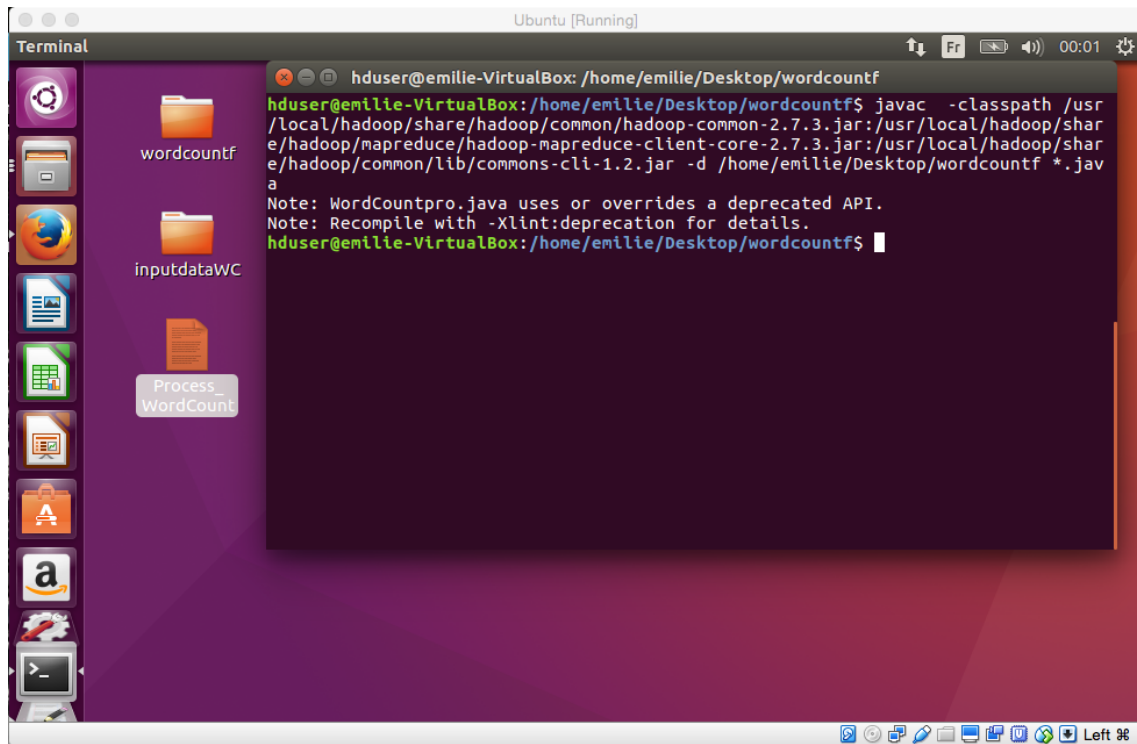


Figure 6: Checking that the data has been imported.

In the **wordcountf** folder on my Desktop, I inserted my MapReduce code for stopwords: **WordCountpro.java**. Placing myself in the path of the folder, I ran the following command (also found in the **Process\_WordCount.txt** file). This allows us to create the different files necessary to create the jar, that we then place in a **wordcountc** file.



The screenshot shows a terminal window titled "Terminal" with the prompt "hduser@emilie-VirtualBox: /home/emilie/Desktop/wordcountf". The command executed is `javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.3.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.7.3.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d /home/emilie/Desktop/wordcountf *.java`. The output shows a note about a deprecated API and a recommendation to recompile with `-Xlint:deprecation`. The terminal window is overlaid on a desktop environment with a purple background and a sidebar containing icons for various applications and files, including "wordcountf", "inputdataWC", and "Process\_WordCount".

```
hduser@emilie-VirtualBox: /home/emilie/Desktop/wordcountf
hduser@emilie-VirtualBox:/home/emilie/Desktop/wordcountf$ javac -classpath /usr
/local/hadoop/share/hadoop/common/hadoop-common-2.7.3.jar:/usr/local/hadoop/shar
e/hadoop/mapreduce/hadoop-mapreduce-client-core-2.7.3.jar:/usr/local/hadoop/shar
e/hadoop/common/lib/commons-cli-1.2.jar -d /home/emilie/Desktop/wordcountf *.jav
a
Note: WordCountpro.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
hduser@emilie-VirtualBox:/home/emilie/Desktop/wordcountf$
```

Figure 7: Creating intermediary files.

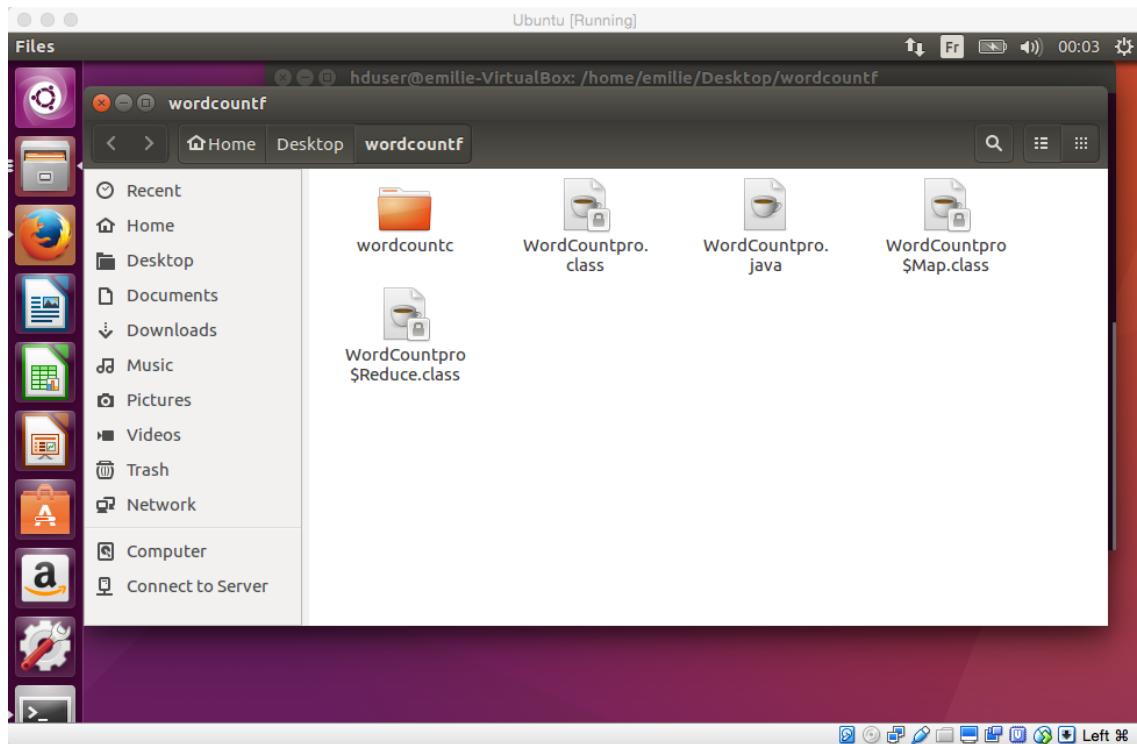


Figure 8: Placing the files in wordcountc.

Then, after having done the compiling, we convert the **wordcountc** folder in jar. As we can see, the operation succeeded.



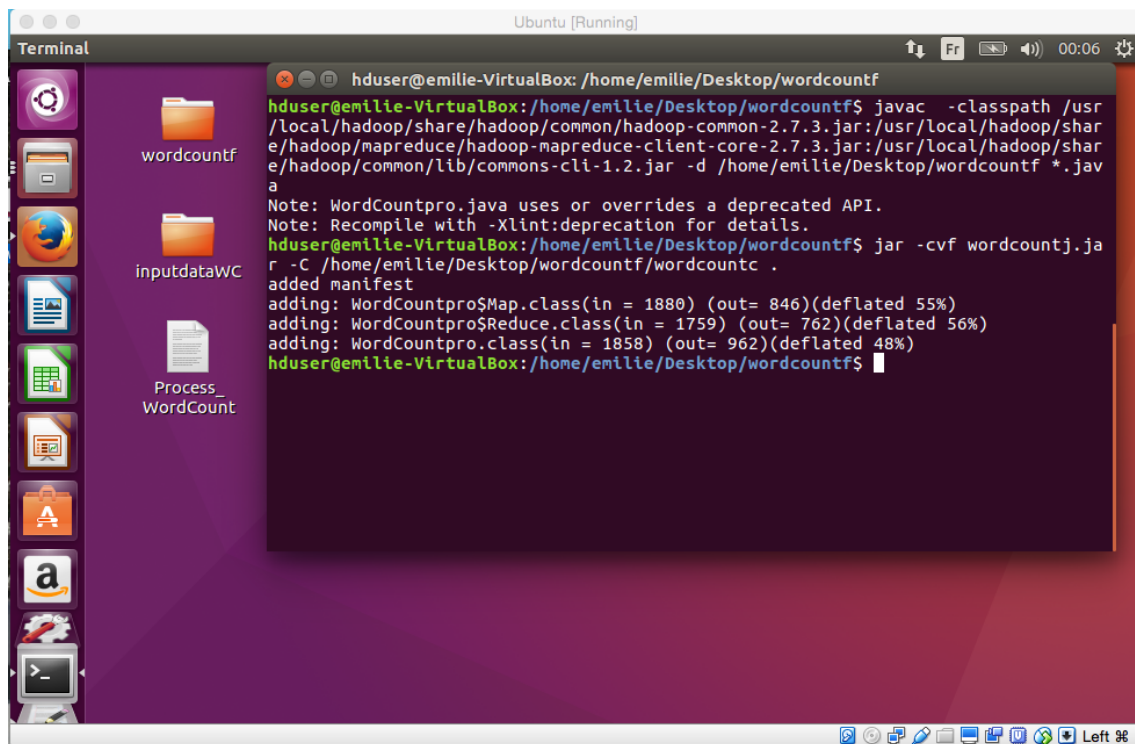


Figure 9: Converting the wordcountc folder to jar.

Now, we will be able to run the stopword MapReduce program (here called WordCountpro) whilst changing various settings.

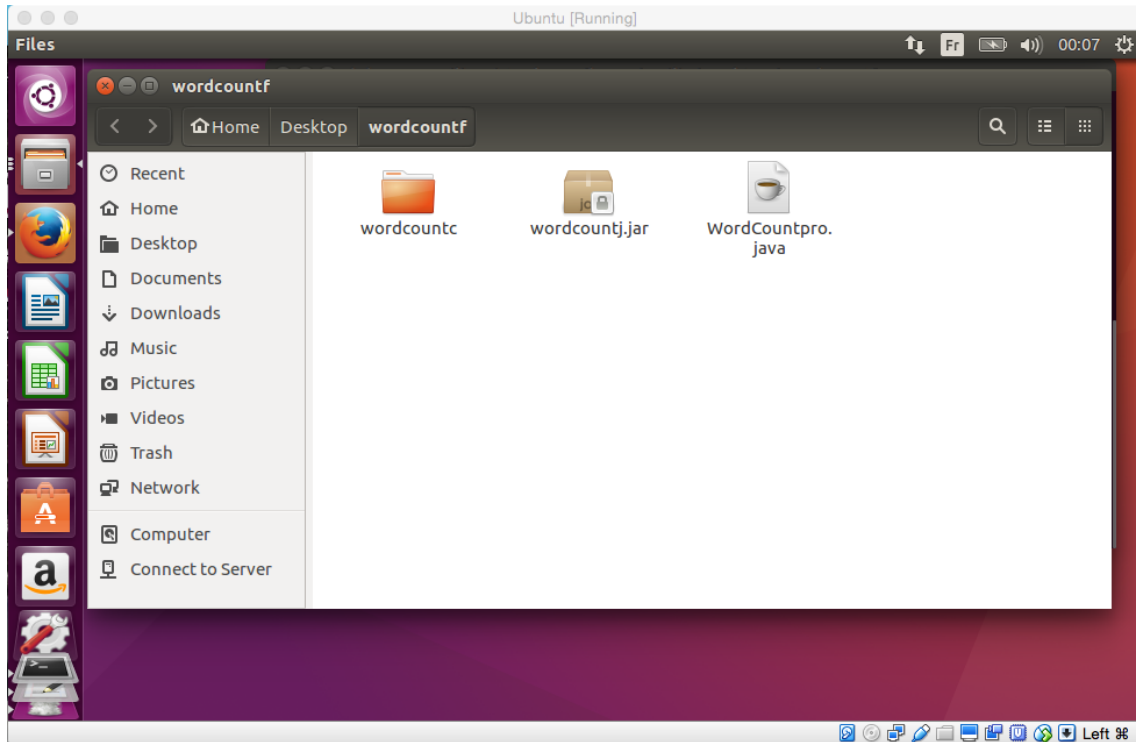


Figure 10: Checking jar creation.

## 2.1 10 reducers and no combiner

In order to run the MapReduce with 10 reducers and no combiner, we run the following command line in `/usr/local/hadoop`:

```
1 bin/hadoop jar /home/emilie/Desktop/wordcountf/wordcountj.jar
  WordCountpro /user/inputdataWC outputwc 10 0 0
```

See the **WordCountpro.java** and the **Process\_WordCount.txt** for full detail. As seen at localhost:50070, the job was a success.

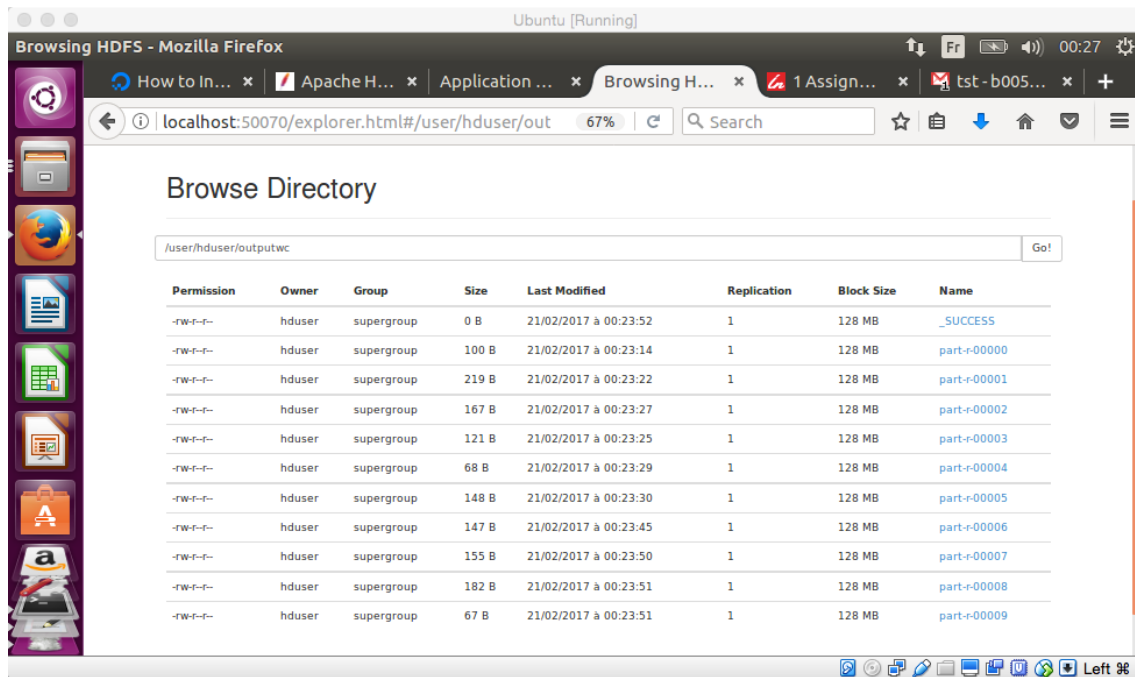


Figure 11: Stopword job with 10 0 0.

And in order to create a full **stopwords.csv**file, we ran the following command lines to merge the files and transform the output in csv:

```
1 cat Downloads/* > Desktop/stopwords.txt
2 sed 's/\t/,/g' Desktop/stopwords.txt > Desktop/stopwords.csv
```

	A	B	C	D	E	F	G	H	I	J	K
1	about	7975									
2	be	28572									
3	before	5715									
4	by	20762									
5	her	28607									
6	much	5395									
7	old	4716									
8	up	10753									
9	where	4718									
10	you	44587									
11	after	4731									
12	been	10350									
13	come	6588									
14	d	10531									
15	down	6182									
16	get	4399									
17	got	4494									
18	have	24606									
19	here	6376									

Figure 12: Extract of the stopwords.csv file.

As we can see at **localhost:8088**, the job took **1min37sec**.

Application application\_1487582138123\_0012 - Mozilla Firefox

localhost:8088/cluster/app/application\_1487582138123\_0012

Application application\_1487582138123\_0012

Cluster

- About
- Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler
- Tools

Kill Application

Application Overview

User: [hduser](#)

Name: WordCountJob

Application Type: MAPREDUCE

Application Tags:

YarnApplicationState: FINISHED

Queue: [default](#)

FinalStatus Reported by AM: SUCCEEDED

Started: Tue Feb 21 00:22:15 +0100 2017

Elapsed: 1mins, 37sec

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: 558845 MB-seconds, 437 vcore-seconds

Show 20 entries

Attempt ID	Started	Node	Logs	Blacklisted Nodes
appattempt_1487582138123_0012_000001	Tue Feb 21 00:22:15 +0100 2017	<a href="#">http://emilie.VirtualBox:8042</a>	<a href="#">Logs</a>	N/A

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Figure 13: 10 0 0 time.

## 2.2 10 reducers and 1 combiner

In order to run the same process with 10 reducers and 1 combiner, we run the following command line, after having deleted the previous output files from the hdfs:

```
1 bin/hadoop jar /home/emilie/Desktop/wordcountf/wordcountj.jar  
   WordCountpro /user/inputdataWC outputwc 10 1 0
```

As we can see, this time the job took **1min14sec**.

The screenshot shows the Hadoop YARN web interface in a Mozilla Firefox browser window. The URL is `localhost:8088/cluster/app/application_1487582138123_0013`. The page title is "Application application\_1487582138123\_0013". The interface includes a sidebar with navigation links like "Cluster", "About", "Nodes", "Node Labels", "Applications", "NEW", "SAVING", "SUBMITTED", "ACCEPTED", "RUNNING", "FINISHED", "FAILED", "KILLED", "Scheduler", and "Tools". The main content area is divided into several sections:

- Kill Application**: A button to kill the application.
- Application Overview**: A table showing application details.

User:	hduser
Name:	WordCountjob
Application Type:	MAPREDUCE
Application Tags:	
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Tue Feb 21 00:48:09 +0100 2017
Elapsed:	1mins, 14sec
Tracking URL:	<a href="#">History</a>
Diagnostics:	
- Application Metrics**: A table showing resource 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:	413701 MB-seconds, 319 vcore-seconds
- Entries**: A table showing application attempts.

Attempt ID	Started	Node	Logs	Blacklisted Nodes
<a href="#">appattempt_1487582138123_0013_000001</a>	Tue Feb 21 00:48:09 +0100 2017	<a href="#">http://emilie-VirtualBox:8042</a>	<a href="#">Logs</a>	N/A

The bottom of the page shows "Showing 1 to 1 of 1 entries" and navigation links: "First", "Previous", "1", "Next", "Last".

Figure 14: 10 1 0 time.

## 2.3 10 reducers and 1 combiner with compression

We proceed the same way with:

```
1 bin/hadoop jar /home/emilie/Desktop/wordcountf/wordcountj.jar  
   WordCountpro /user/inputdataWC outputwc 10 1 1
```

And the output is of **1min17sec**.

The screenshot shows the Hadoop YARN web interface in a Mozilla Firefox browser window. The URL is `localhost:8088/cluster/app/application_1487582138123_0014`. The page title is "Application application\_1487582138123\_0014". The interface is logged in as "dt:who".

On the left sidebar, there is a "Cluster" section with links for "About", "Nodes", "Node Labels", and "Applications". The "Applications" link is selected, showing a list of application states: NEW, SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, and KILLED. The "Scheduler" link is also visible.

The main content area is titled "Kill Application" and "Application Overview". It displays the following details:

- User: hduser
- Name: WordCountjob
- Application Type: MAPREDUCE
- Application Tags:
- YarnApplicationState: FINISHED
- Queue: default
- FinalStatus Reported by AM: SUCCEEDED
- Started: Tue Feb 21 00:53:12 +0100 2017
- Elapsed: 1mins, 17sec
- Tracking URL: [History](#)
- Diagnostics:

Below the overview, there is an "Application Metrics" section showing the following data:

- 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: 427904 MB-seconds, 328 vcore-seconds

At the bottom, there is a table with columns: Attempt ID, Started, Node, Logs, and Blacklisted Nodes. The table shows one entry:

Attempt ID	Started	Node	Logs	Blacklisted Nodes
<a href="#">appattempt_1487582138123_0014_000001</a>	Tue Feb 21 00:53:12 +0100 2017	<a href="#">http://emilie-VirtualBox:8042</a>	<a href="#">Logs</a>	N/A

The table indicates "Showing 1 to 1 of 1 entries". Navigation links for "First", "Previous", "Next", and "Last" are provided.

Figure 15: 10 1 1 time.

## 2.4 50 reducers and 1 combiner with compression

Finally, with 50 reducers, a combiner and compression, we ran:

```
1 bin/hadoop jar /home/emilie/Desktop/wordcountf/wordcountj.jar  
   WordCountpro /user/inputdataWC outputwc 50 1 1
```

And the final output was of **4min17sec**.

The screenshot shows the Hadoop web interface in a Mozilla Firefox browser window. The URL is `localhost:8088/cluster/app/application_1487582138123_0015`. The page title is "Application application\_1487582138123\_0015". The interface includes a sidebar with navigation links like "Cluster", "About", "Nodes", "Node Labels", "Applications", "NEW", "SAVING", "SUBMITTED", "ACCEPTED", "RUNNING", "FINISHED", "FAILED", "KILLED", and "Scheduler". The main content area is divided into several sections:

- Kill Application**: A button to kill the application.
- Application Overview**: A table showing application details.

Property	Value
User	hduser
Name	WordCountjob
Application Type	MAPREDUCE
Application Tags	
YarnApplicationState	FINISHED
Queue	default
FinalStatus Reported by AM	SUCCEEDED
Started	Tue Feb 21 00:57:16 +0100 2017
Elapsed	4mins, 17sec
Tracking URL	History
Diagnostics	
- Application Metrics**: A table showing resource metrics.

Metric	Value
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	1930395 MB-seconds, 1597 vcore-seconds
- Attempt List**: A table showing application attempts.

Attempt ID	Started	Node	Logs	Blacklisted Nodes
<a href="#">appattempt_1487582138123_0015_000001</a>	Tue Feb 21 00:57:16 +0100 2017	<a href="#">http://emilie-VirtualBox:8042</a>	<a href="#">Logs</a>	N/A

The bottom of the page shows "Showing 1 to 1 of 1 entries" and navigation links: "First", "Previous", "1", "Next", "Last".

Figure 16: 50 1 1 time.



To conclude:

- Adding a combiner reduces the computation time. This is because the combiner reduces the output pairs of each mapper and thus lowers the load for the reducers.
- Adding compression increases the computation time compared to simply having 10 reducers and a combiner. Compression mainly reduces network transfer times, so it is not useful on a single node cluster.
- Going up to 50 reducers largely increases the computation time. Since we are splitting the output in too many ways, each mapper has to save its output for all the forthcoming reducers and thus reduces performance.

### 3 Simple and complex inverted index

We proceeded as before and managed to obtain the following results.

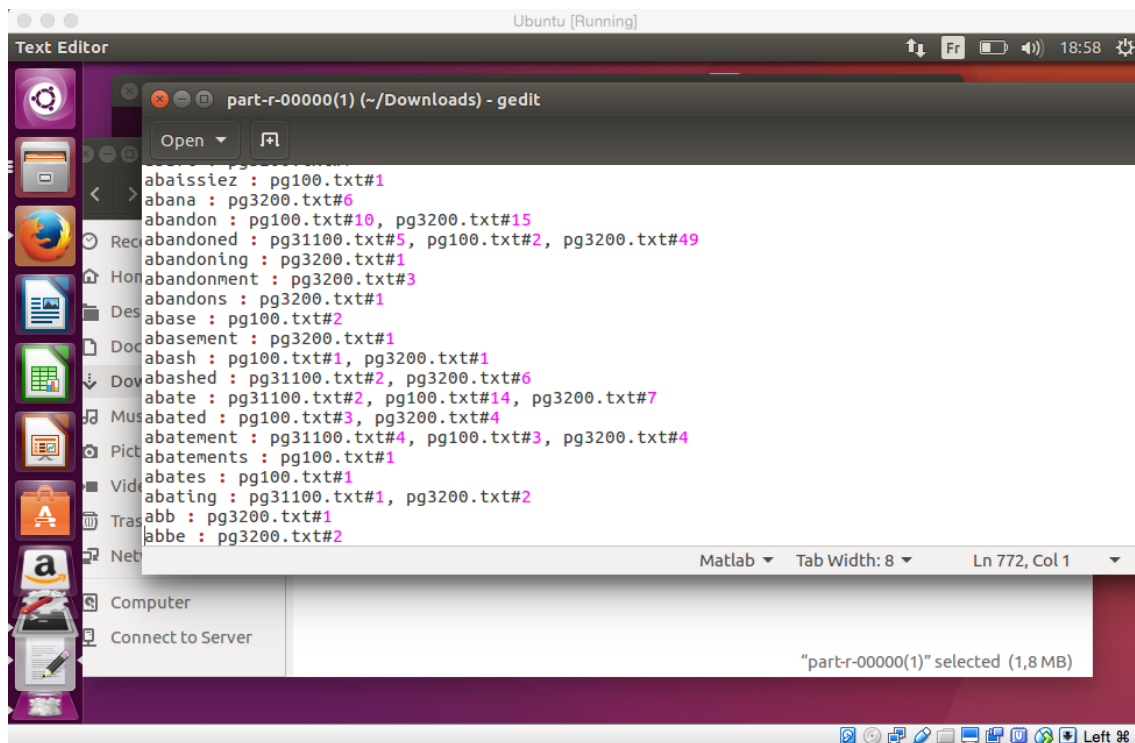
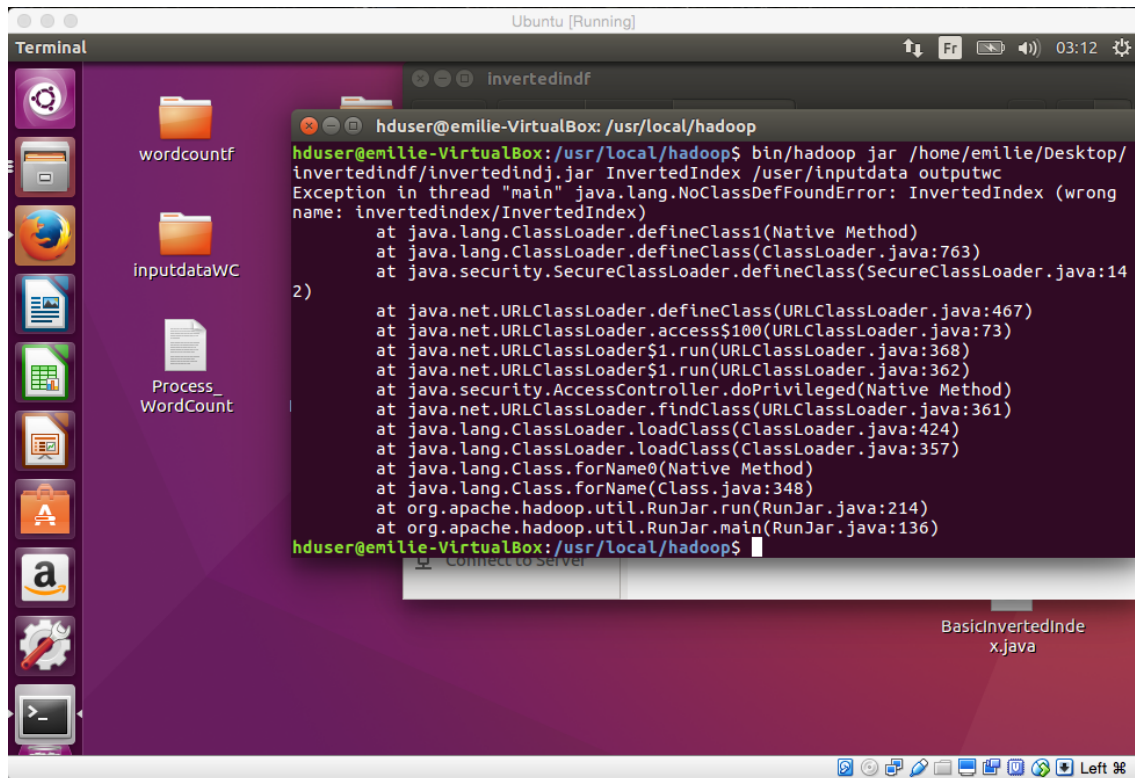


Figure 17: Good results obtained.

However, a bug is now occurring and, after numerous attempts to correct it, we are stuck with the following error:



The screenshot shows a terminal window titled 'Terminal' with the prompt 'hduser@emilie-VirtualBox: /usr/local/hadoop'. The user has executed the command: `bin/hadoop jar /home/emilie/Desktop/invertedindf/invertedindj.jar InvertedIndex /user/inputdata outputwc`. The output shows an exception: `Exception in thread "main" java.lang.NoClassDefFoundError: InvertedIndex (wrong name: invertedindex/InvertedIndex)`. The stack trace includes the following frames: `java.lang.ClassLoader.defineClass1(Native Method)`, `java.lang.ClassLoader.defineClass(ClassLoader.java:763)`, `java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)`, `java.net.URLClassLoader.defineClass(URLClassLoader.java:467)`, `java.net.URLClassLoader.access$100(URLClassLoader.java:73)`, `java.net.URLClassLoader$1.run(URLClassLoader.java:368)`, `java.net.URLClassLoader$1.run(URLClassLoader.java:362)`, `java.security.AccessController.doPrivileged(Native Method)`, `java.net.URLClassLoader.findClass(URLClassLoader.java:361)`, `java.lang.ClassLoader.loadClass(ClassLoader.java:424)`, `java.lang.ClassLoader.loadClass(ClassLoader.java:357)`, `java.lang.Class.forName0(Native Method)`, `java.lang.Class.forName(Class.java:348)`, `org.apache.hadoop.util.RunJar.run(RunJar.java:214)`, and `org.apache.hadoop.util.RunJar.main(RunJar.java:136)`. The terminal window is overlaid on a desktop environment with a purple background and various icons. A file named 'BasicInvertedIndex.java' is visible in the bottom right corner of the terminal window.

```
hduser@emilie-VirtualBox: /usr/local/hadoop
hduser@emilie-VirtualBox:/usr/local/hadoop$ bin/hadoop jar /home/emilie/Desktop/
invertedindf/invertedindj.jar InvertedIndex /user/inputdata outputwc
Exception in thread "main" java.lang.NoClassDefFoundError: InvertedIndex (wrong
name: invertedindex/InvertedIndex)
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:763)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:14
2)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:467)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:73)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:368)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:362)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:361)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:348)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:214)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
hduser@emilie-VirtualBox: /usr/local/hadoop$
```

Figure 18: Error currently obtained.

## 4 Number of unique words

In order to do so, we have to create a counter such as:

```
1 public static enum CUSTOM_COUNTER {  
2     UNIQUE_WORDS,  
3 };
```

But we did not manage to make it work.