**Exercise 6**

*Get started with Hadoop*

**Prior Knowledge**

Unix Command Line Shell

Simple Python

**Learning Objectives**

Understand how to start and stop HDFS, and transfer files into and out of the Hadoop filesystem.

Understand how to create a Python mapper/reducer and execute this on a single-node setup of Hadoop.

Stop and start YARN.

**Software Requirements**

(see separate document for installation of these)

* Apache Hadoop 2.7.1
* Python 2.7.x
* Nano text editor or other text editor

**Part A: Hadoop File System (HDFS)**

1. Make sure you are running the Ubuntu VM, and start a fresh terminal window.
2. We need to clean up the Hadoop filesystem and re-format.  
   First, make sure the hadoop fs is stopped:  
     
   stop-dfs.sh
3. Now let’s empty the HDFS storage directory.  
     
   rm -rf /app/hadoop/tmp/dfs/

1. Now let’s format the HDFS filesystem:  
   hadoop namenode -format  
     
   You should see a lot of output ending something similar to this:  
   It may prompt you   
     
   Re-format filesystem in Storage Directory /app/hadoop/tmp/dfs/name ? (Y or N)   
   

Choose **y**

1. Now let’s start the Hadoop filesystem. Type:   
     
   start-dfs.sh  
     
   You should see output like:



1. Now let’s make a directory:  
     
   hadoop fs -mkdir -p /user/oxclo/wind
2. And check it worked:

hadoop fs -ls -R /

You should see:

15/10/22 09:24:48 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:36 /user

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:36 /user/oxclo

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:36 /user/oxclo/wind

1. Now let’s copy some datafiles from the local filesystem into the HDFS:  
     
   hadoop fs -put ~/datafiles/wind/2015/\* /user/oxclo/wind

If you repeat the command to list the files on the hadoop filesystem:

hadoop fs -ls -R /

you should see the new files in place:  
15/10/22 09:45:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:36 /user

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:36 /user/oxclo

drwxr-xr-x - oxclo supergroup 0 2015-11-11 16:37 /user/oxclo/wind

-rw-r--r-- 1 oxclo supergroup 2716705 2015-11-11 16:37 /user/oxclo/wind/SF04.csv

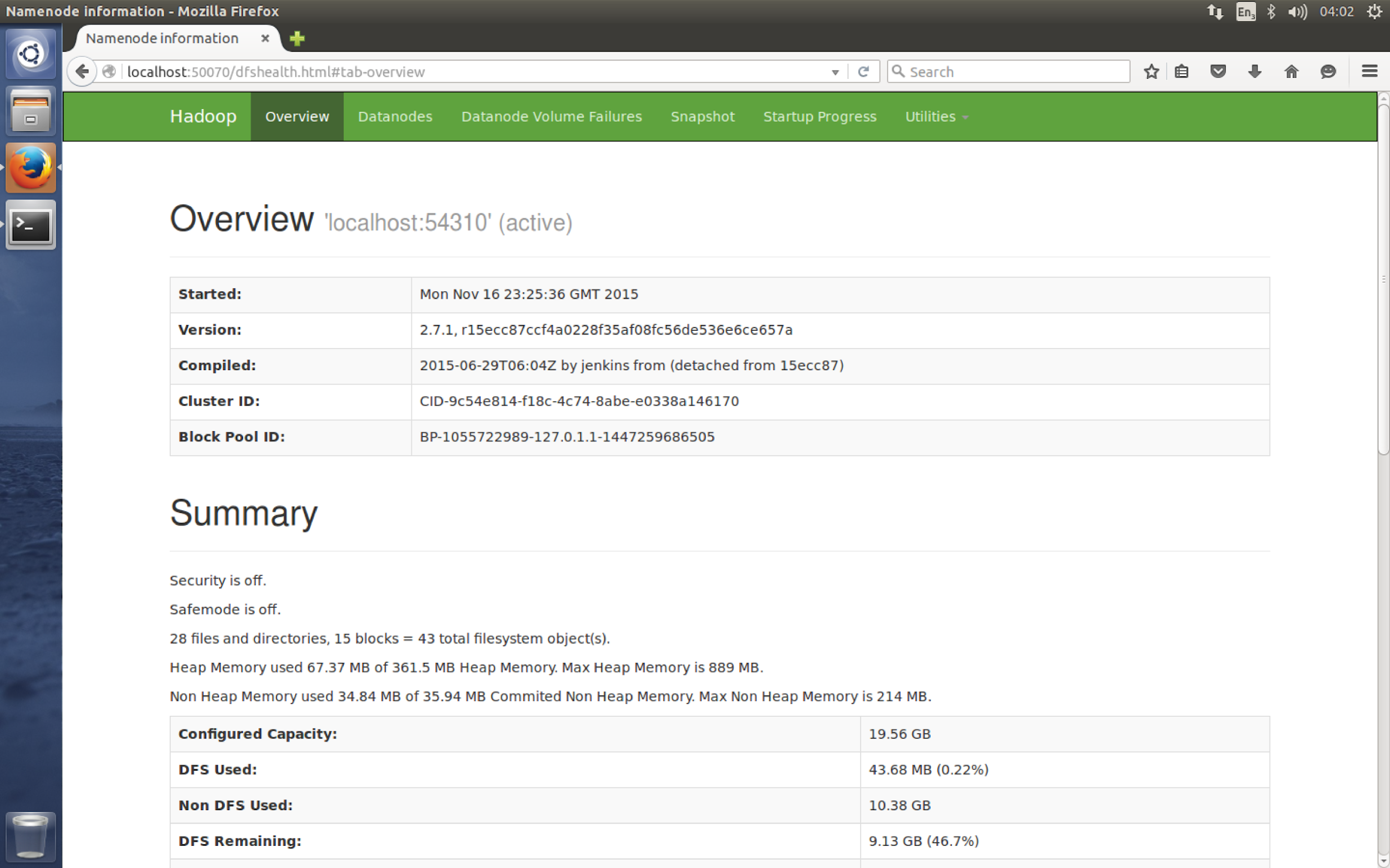
-rw-r--r-- 1 oxclo supergroup 7896961 2015-11-11 16:37 /user/oxclo/wind/SF15.csv

-rw-r--r-- 1 oxclo supergroup 7250586 2015-11-11 16:37 /user/oxclo/wind/SF17.csv

-rw-r--r-- 1 oxclo supergroup 5538010 2015-11-11 16:37 /user/oxclo/wind/SF18.csv

-rw-r--r-- 1 oxclo supergroup 5818496 2015-11-11 16:37 /user/oxclo/wind/SF36.csv

-rw-r--r-- 1 oxclo supergroup 4952077 2015-11-11 16:37 /user/oxclo/wind/SF37.csv

1. You can browse the HDFS Web UI by going to <http://localhost:50070>  
   You will see something like:
2.   
   Take a look at the various tabs.
3. Congratulations you have successfully completed part A.

**Part B. Creating a mapper and reducer, and running map reduce**

1. We have a lot of environmental and wind data from San Francisco, which we can analyse for various information.
2. First, let’s look at the format of the input data.  
   Type   
   hadoop fs -cat /user/oxclo/wind/SF37.csv | head  
   You should see:  
     
   
3. There is a header line and then a set of data, all in comma separated value (CSV) format.
4. Let’s start by calculating maximum wind velocity for each district over the given period (the current year).
5. To do this, we can simply extract the Station\_ID and the Wind\_Velocity\_Mtr\_sec from each line using the mapper, and then we can summarize this information in the Reducer.
6. There is a skeleton mapper written for you at the URL:  
   <http://freo.me/oxclo-mapper>
7. Create a directory for your code:  
   mkdir ~/wind-analysis  
   cd ~/wind-analysis
8. Edit this mapper to output the right key/value pairs.
   1. You can use any Unix text editor. If you know how, then nano is a good terminal based editor.   
        
      I have installed Sublime Text (subl on the command line) which is a good Python editor.  
      You can use subl like this:  
      subl wind-mapper.py

Some people like PyCharm and you can install that if you prefer

* 1. The usual way of passing the key/value pairs from the mapper to the reducer is to use Tab delimited lines. A quick and easy way of printing those out in Python is to create an array of results and then use the join function to link them up with tabs:

result = [ k, v ]  
print ('\t'.join(result))

* 1. Don’t forget that Python is tab/space sensitive!
  2. You may need to cast velocity as a string using str(velocity)

1. Before we run this using Hadoop, we can simply test our script using the command-line.   
   1. We first need to make the mapper.py executable:  
      chmod +x ./wind-mapper.py
   2. We will use the local copy of one of the files
   3. Type:   
      cat ~/datafiles/wind/2015/SF37.csv | ./wind-mapper.py
   4. You should see many data lines like this printed:  
      
2. Now we need to create a reducer:  
   1. The first task is to find the maximum wind speed recorded for each station.
   2. Once again start with the skeleton that is provided in:  
      <http://freo.me/oxclo-reducer>
   3. Create a file locally and edit it to work. Call it wind-reducer.py  
      Don’t forget to   
      chmod +x ./wind-reducer.py
   4. I recommend using a python dictionary (<https://docs.python.org/2/tutorial/datastructures.html#dictionaries>).

to collect the maximum value by station id.

* 1. Hint: you need to convert the number to a string before printing it with the join function, you can do this by using str(max)

1. Check that your code works locally before we try it with Hadoop  
   cat ~/datafiles/wind/2015/SF37.csv \  
   | ./wind-mapper.py | ./wind-reducer.py
2. You should see:   
   SF37 7.079  
   printed out.
3. Now we are ready to run this as a map-reduce job using Hadoop. Firstly, this will run different files on (potentially) different systems and processes. Secondly, it will access the data from HDFS instead of the local file system.
4. Before we can run the map-reduce job, we need to start up YARN, the job scheduler.
   1. Type:

start-yarn.sh

* 1. You should see:  
       
     starting yarn daemons

starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-oxclo.out

localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-oxclo.out

1. Now we can initiate the job. To run the python job, we utilize a generic capability of Hadoop to run code that uses standard input/output. Of course we’ve written those Python programs to do exactly that. This capability is called Hadoop Streaming  
   1. Type (on one line)  
        
      yarn jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.3.jar -input /user/oxclo/wind/ -output /user/oxclo/output -mapper ./wind-mapper.py -reducer ./wind-reducer.py
   2. You should see a **lot** of log output, ending something *similar* to:  
      
   3. Now you can check if there is any output:  
        
      hadoop fs -cat /user/oxclo/output/part-00000
   4. You should see:  
      15/10/23 14:38:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

SF04 34.12

SF36 11.05

SF15 7.92

SF17 5.767

SF18 10.57

SF37 7.079

1. If you do not see the same output, then check your code and re-run. If you re-run you will need to specify a new output directory (e.g. output-2).
2. Now modify your code to calculate the average wind velocity at each station.   
     
   Hint 1: copy your reducer.py code to reducer-average.py to start with.  
   Hint 2: you shouldn’t need to change the mapper  
   Hint 3: you will need to specify a new output directory  
     
   Run your code using hadoop and compare your answers to these:  
     
   SF04 2.30098174812

SF36 2.46417253091

SF15 1.82141456775

SF17 0.518350025349

SF18 2.22022343917

SF37 2.2604035055

1. Congratulations! You have completed this exercise.
2. If you want to see a sample of code that calculates both the maxes and the averages, it is here:  
   <https://github.com/pzfreo/ox-clo/tree/master/code/wind-analysis/complete>
3. **Extension:**

Determine which direction (N,NE,E,SE,S,SW,W,NW) has had the strongest average winds this year across all stations.