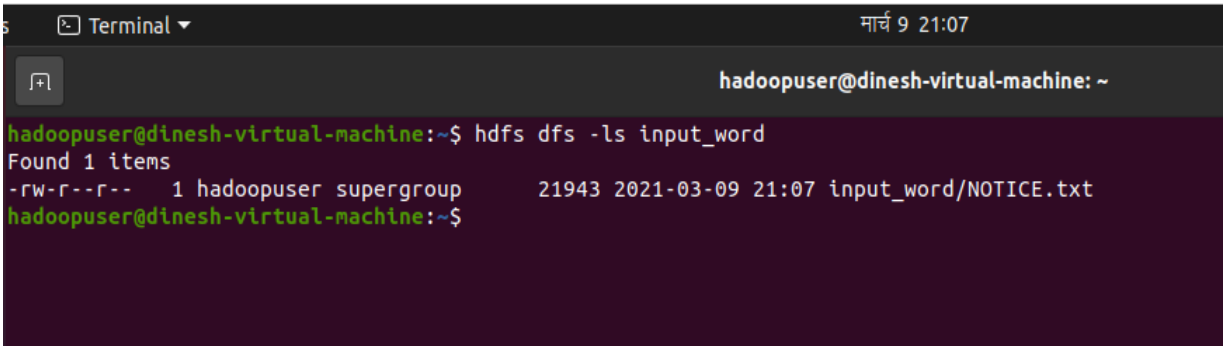
Running Map reduce in python

1) Python3 comes preinstalled in newer version of Ubuntu. However, if you want to check which version is installed on your machine:
   - For python3 version check : **pyhthon3 - -version**

     OR
   - For python2 version check: **python2 - -version**
2) If the python is not installed then you can install the python with the following code

```
sudo apt install python3
```

3) Next we have to create a folder and insert any local file in the HDFS.



4) Now we have to create a mapper and reducer. To create a mapper class first we have to create a empty file and add the code related.
5) Type " **touch mapper.py**" and press enter. Similarly for reducer "**touch reducer**.py" and press enter. You will have two file located in your **/home/hdoop/** directory.
6) Open the mapper.py by typing "**nano mapper.py**" and add the flowing code

```
#!/usr/bin/env python3

"""mapper.py"""


import sys
```

```
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print ('%s\t%s' % (word, 1))
```

NOTE: I have python3 in my machine so I have changed that accordingly. In python 3 we have to add parenthesis in print statement and remove in python version 2

Example:

- For python version 2 = print '%s\t%s' % (word, 1)
- For python version 3 = print ('%s\t%s' % (word, 1))

```python
#!/usr/bin/env python3
"""mapper.py"""

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print ('%s\t%s' % (word, 1))
```

7) Now open the reducer file by typing “reducer.py” and press enter. add the following code in reducer.py

```
es    Terminal ▾
                                                    hadoo
hadoopuser@dinesh-virtual-machine:~$ touch reducer.py
```

```
#!/usr/bin/env python3

"""reducer.py"""


from operator import itemgetter
```

```
import sys


current_word = None
current_count = 0
word = None


# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()


    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)


    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue


    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
     current_count += count
    else:
        if current_word:
            # write result to STDOUT
```

```
        print ('%s\t%s' % (current_word, current_count))

    current_count = count

    current_word = word


# do not forget to output the last word if needed!

if current_word == word:

  print ('%s\t%s' % (current_word, current_count))
```

```
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print ('%s\t%s' % (current_word, current_count))
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print ('%s\t%s' % (current_word, current_count))
```

8) Now we have to give permission for the mapper and reducer. Type the following stepwise step.
   - `chmod +x /home/hduser/mapper.py`
   - `chmod +x /home/hduser/reducer.py`

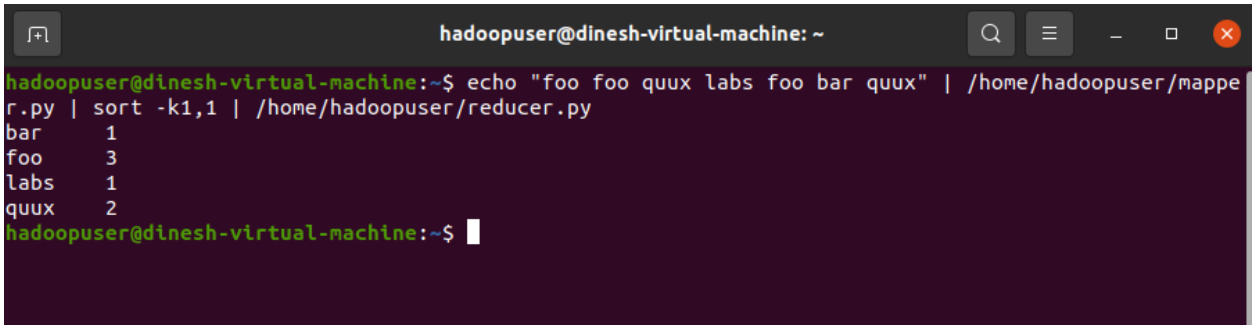
Testing the mapper.py and reducer.py

9) To test the mapper type the following code and enter.

```
echo "foo foo quux labs foo bar quux" | /home/hdoopuser/mapper.py
```

10) Now test the mapper and reducer code together

```
echo "foo foo quux labs foo bar quux" | /home/hdoopuser/mapper.py | sort -k1,1 |
/home/hdoopduser/reducer.py
```



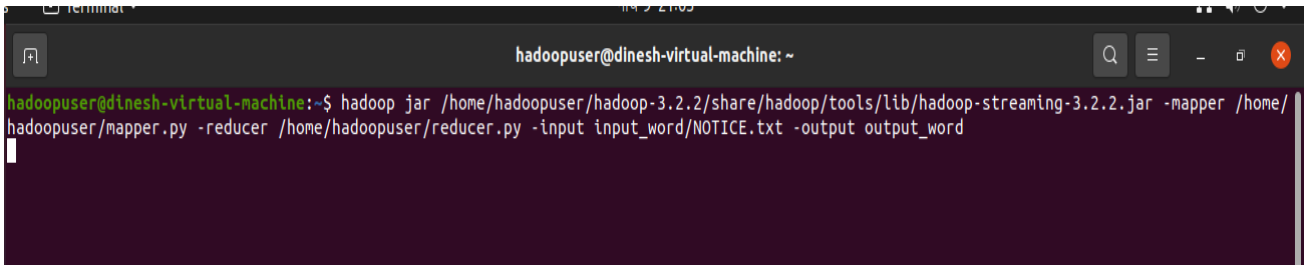
Running the Mapper.py and reducer.py in HDFS

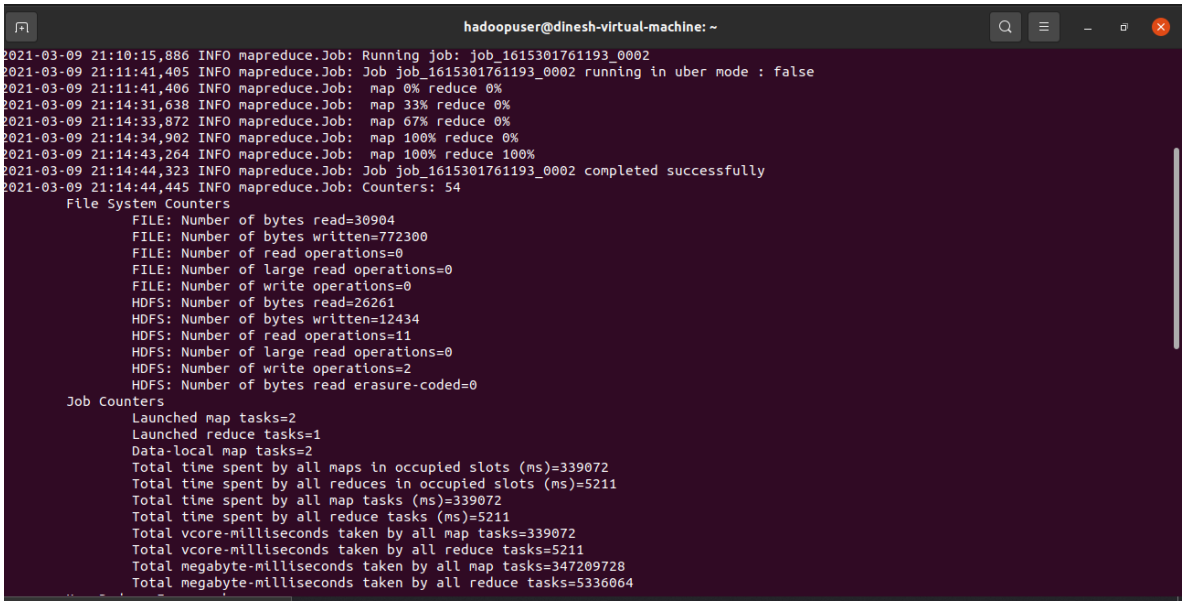
11) Now to execute the code run the following code.

```
hadoop jar /home/hdoop/Hadoop-3.2.2/share/hadoop/tools/lib/hadoop-streaming-3.2.2.jar
-mapper /home/hdoop/mapper.py -reducer /home/hdooppuser/reducer.py -input
input_word/NOTICE.txt -output output_word
```

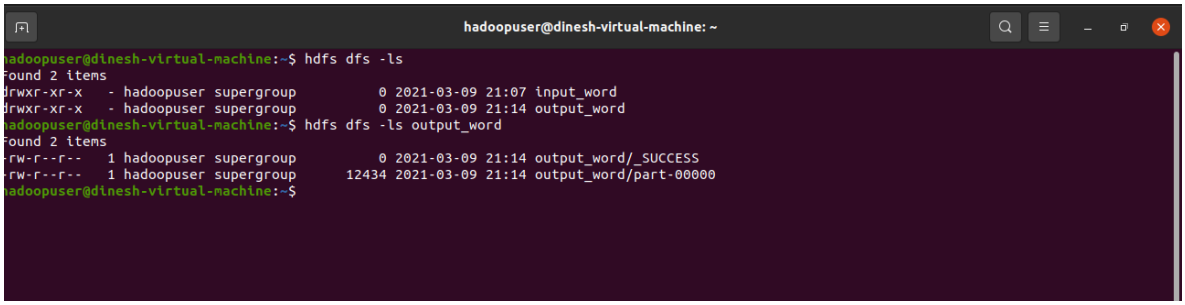
12) if you have output folder then you have to delete that

Note: ***It is just the syntax. Please enter proper credentials as your system has been configured.***



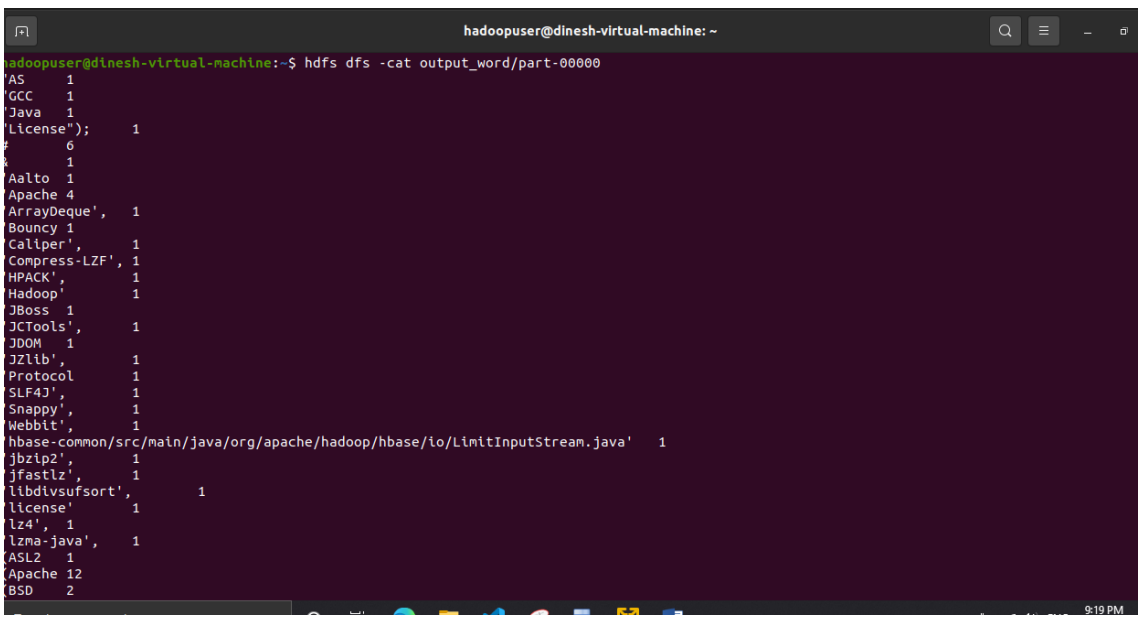


13) Now you will see output folder and the output file. Output word in my case as I have given that specific name while executing map reduce code.



14) Now you can test if the map reduce program in the python worked correctly or not by following command.

```
hdfs dfs -cat output_word/part-00000
```



!!WELL DONE!!

15) References

- Writing An Hadoop MapReduce Program In Python (michael-noll.com)