CSE 587 Lab 4 Readme File

Activity 1: WordCount on tweets

- There are 5 folders present in the CSE587Lab4 folder. In that go to ‘notebooks’.
- Among them Word_Cloud1.ipynb and Word_Cloud2.ipynb are required in order to visualize wordcloud. Word_Cloud1.ipynb was used for extracting and cleaning the data which is required for generating the wordcloud.
- After tweets have been extracted and cleaned using this notebook they are to be fed to the MapReduce program.
- The extracted tweets after Word_Cloud1.ipynb are stored in inputs/activity1/ folder which have to be fed to the MapReduce program.

Commands to execute Activity1

```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity1/ ~/input
hadoop com.sun.tools.javac.Main WordCount.java
jar cf wc.jar WordCount*.class
hadoop jar wc.jar WordCount ~/input/activity1 ~/output1
```

The jar file for this part is (wc.jar)

- This gives us a wordcount for the tweets as output which is stored in outputs/output1 folder.
- This output file is referenced in the second notebook Word_Cloud2.ipynb for generating WordCloud.

Activity 2: Word co-occurence on tweets

- In this part we have prepared our data using the Co_occurence.ipynb notebook in the notebooks folder.
- This data is stored in inputs/activity2/ folder.
- We have to feed this data to the MapReduce program which is done using the following commands. There are two jars for this part (2a.jar which is used for pairs and 2b.jar for stripes)

Commands to execute Activity 2a

```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity2/ ~/input
hadoop com.sun.tools.javac.Main Activity2a.java
jar cf 2a.jar Activity2a*.class
hadoop jar 2a.jar Activity2a ~/input/activity2 ~/output2a
```

The jar for this program is (2a.jar)

Commands to execute Activity 2b

```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity2/ ~/input
hadoop com.sun.tools.javac.Main Activity2b.java
jar cf 2b.jar Activity2b*.class
hadoop jar 2b.jar Activity2b ~/input/activity2 ~/output2b
```

The jar for this program is (2b.jar)

The outputs generated for both pairs and stripes are stored in outputs/output2a and outputs/output2b folders respectively.

Activity 3: WordCount on Classic Latin Text

- In order to run this Activity first paste new_lemmatizer.csv which is present in /notebooks folder at the location “/home/hadoop/” that is the home direcotry in the VM.
- This is because the code would be referencing the path “/home/hadoop/new_lemmatizer.csv” in order to read and store the file in memory.
- The input files are located in /inputs/activity3/ folder

Commands to execute Activity 3

```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity3/ ~/input
hadoop com.sun.tools.javac.Main Activity3.java
jar cf 3.jar Activity3*.class
hadoop jar 3.jar Activity3 ~/input/activity3 ~/output3
```

Output file for this program is stored in outputs/output3/ folder.
The jar for this program is (3.jar)

Activity 4: Word co-occurrence among multiple documents

- In order to run this Activity first paste new_lemmatizer.csv which is present in /notebooks folder at the location “/home/hadoop/” that is the home direcotry.
- This is because the code would be referencing the path “/home/hadoop/new_lemmatizer.csv” in order to read and store the file in memory.
- The input files are located in /inputs/activity4/ folder

Commands to execute Activity 4a

```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity4/ ~/input
hadoop com.sun.tools.javac.Main Activity4a.java
jar cf 4a.jar Activity4a*.class
hadoop jar 4a.jar Activity4a ~/input/activity4 ~/output4a
```

Output file for this program is stored in outputs/output4a/ folder.
The jar for this program is (4a.jar)

Commands to execute Activity 4b

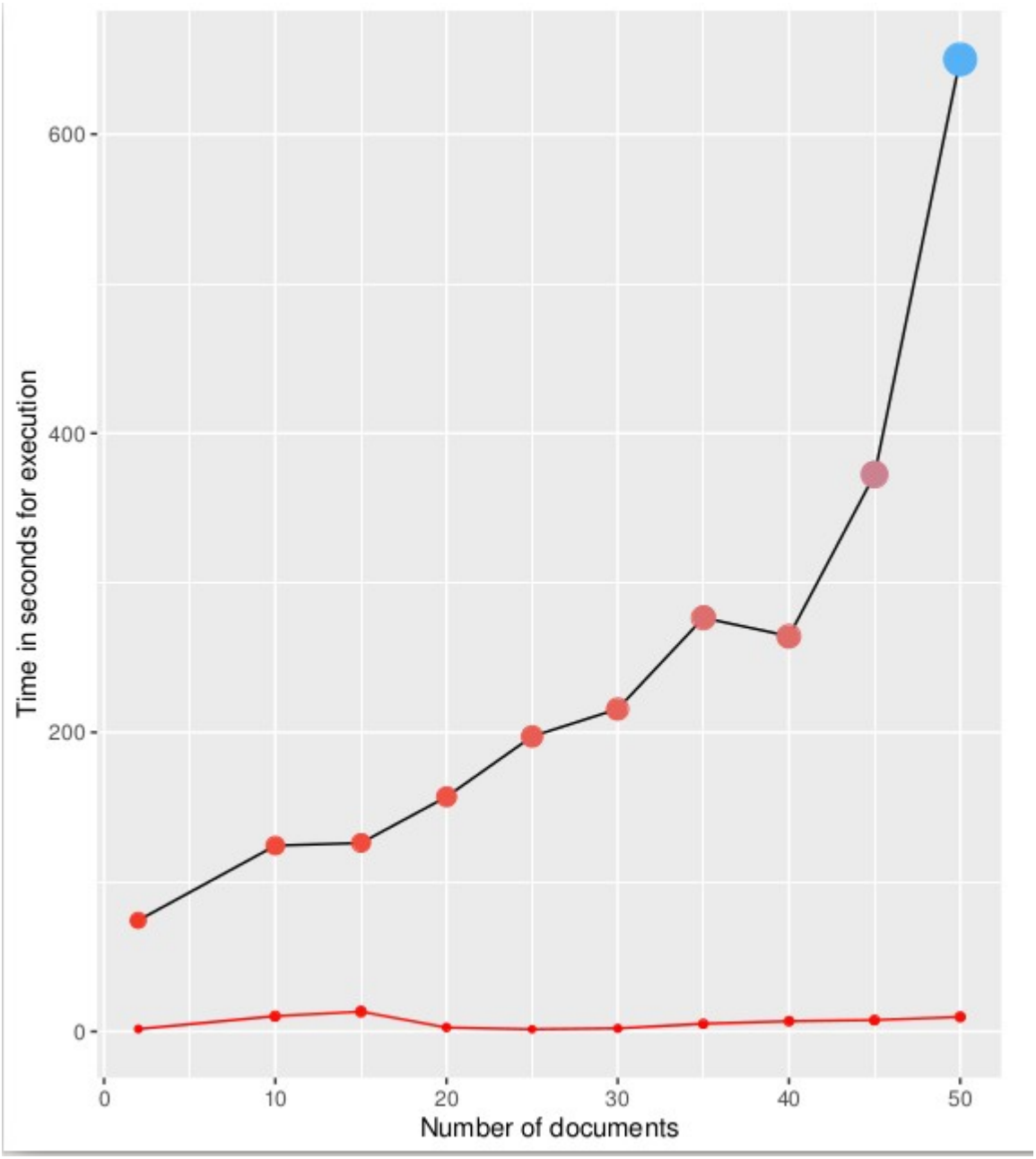
```
hdfs dfs -mkdir -p ~/input/
hdfs dfs -put ~/activity4/ ~/input
hadoop com.sun.tools.javac.Main Activity4b.java
jar cf 4b.jar Activity4b*.class
hadoop jar 4b.jar Activity4b ~/input/activity4 ~/output4b
```

Output file for this program is stored in outputs/output4b/ folder.
The jar for this program is (4b.jar)

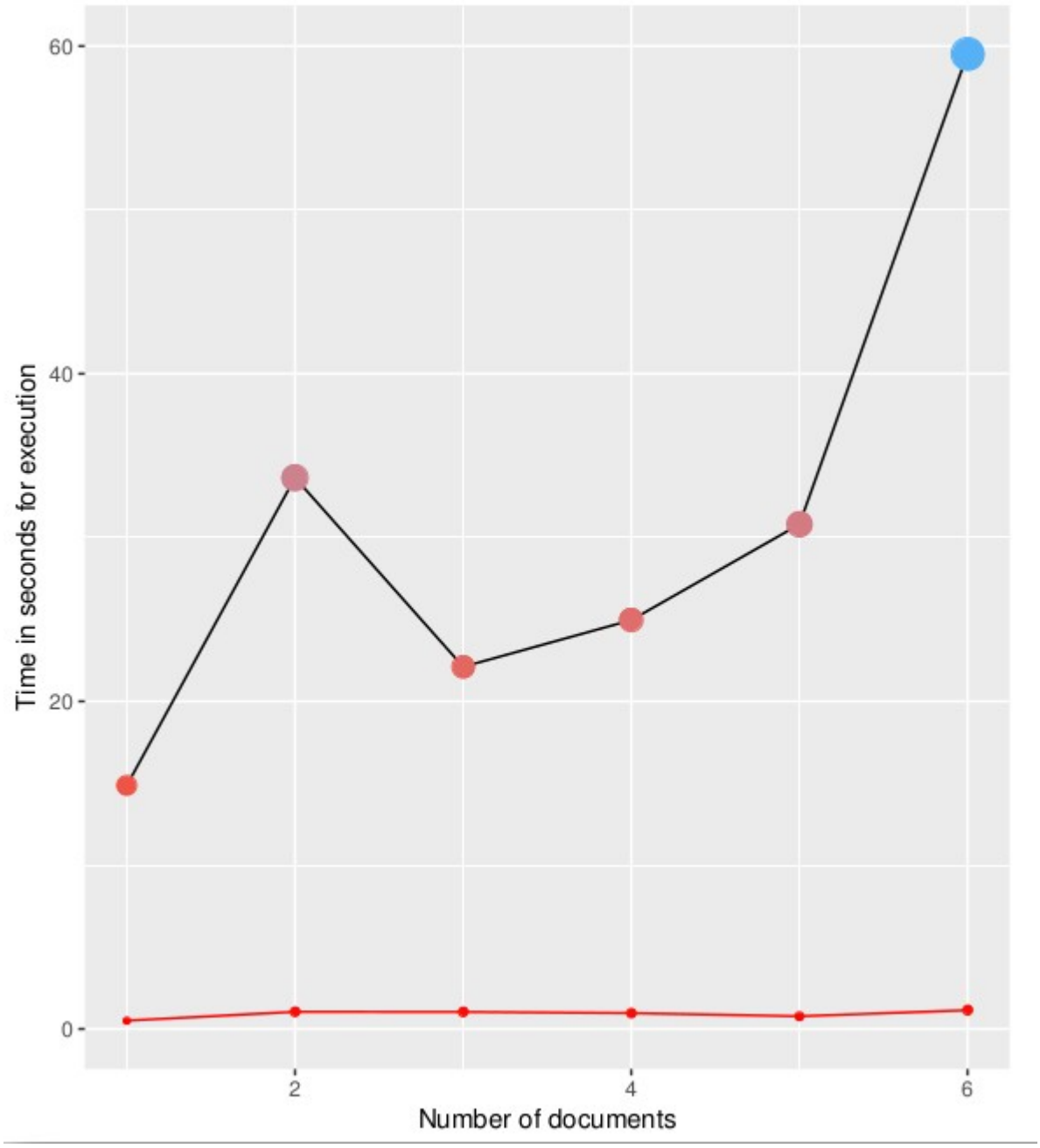
Scaling 2grams and 3grams to multiple documents

The x axis here refers to the number of documents incrementally processed and the y-axis refers to time in seconds.

1. Activity4a: Bi-grams



2. Activity4b : 3 – grams



Conclusion and Inference:

- The black line on the plot indicates elapsed real (wall clock) time used by the process, in seconds.
- The red line on the plot indicates total number of CPU-seconds used by the system on behalf of the process (in kernel mode), in seconds.
- In case of 2-grams it was possible to run the program for up to 50 documents but in case of 3 – grams the performance is slow. Hence it has been scaled up to only 6 documents.
- This is mainly because 3 – grams requires more computing than 2 grams as more combinations have to be considered.
- From the above graphs we can clearly see that the real time significantly increases as we increase the documents however the total number of CPU-seconds used by the system on behalf of the process don't deviate much.

Performance for bi – grams and 3 – grams where,

- **real** - elapsed real (wall clock) time used by the process, in seconds.
- **sys** - total number of CPU-seconds used by the system on behalf of the process (in kernel mode), in seconds.
- **num** – number of documents processed.

real	sys	num
74.345	1.684	2
124.324	10.320	10
126.055	13.272	15
156.942	2.620	20
197.316	1.496	25
215.670	2.116	30
276.552	5.228	35
264.342	6.868	40
372.359	7.664	45
650.059	9.764	50

Bi – grams

real	sys	num
14.872	0.508	1
33.652	1.048	2
22.110	1.040	3
24.962	0.964	4
30.808	0.776	5
59.530	1.144	6

3-grams

Folder Contents

- **inputs** – Contains inputs corresponding to the 4 activities.
- **jars** - Contains jars for all 4 activities.
- **notebooks** – Contains notebooks for preparing data for Activity 1 and Activity2. Also contains ‘new_lemmatizer.csv’ which is to be moved to ‘/home/hadoop’ directory in the VM for executing Activity3 and Activity4.
- **outputs** - Contains outputs corresponding to all 4 activities.
- **sourcecode** - Contains java source code for all the activities.