

# LAB REPORT

## Exercise-2

*By*

Ambuj Mishra

202116003

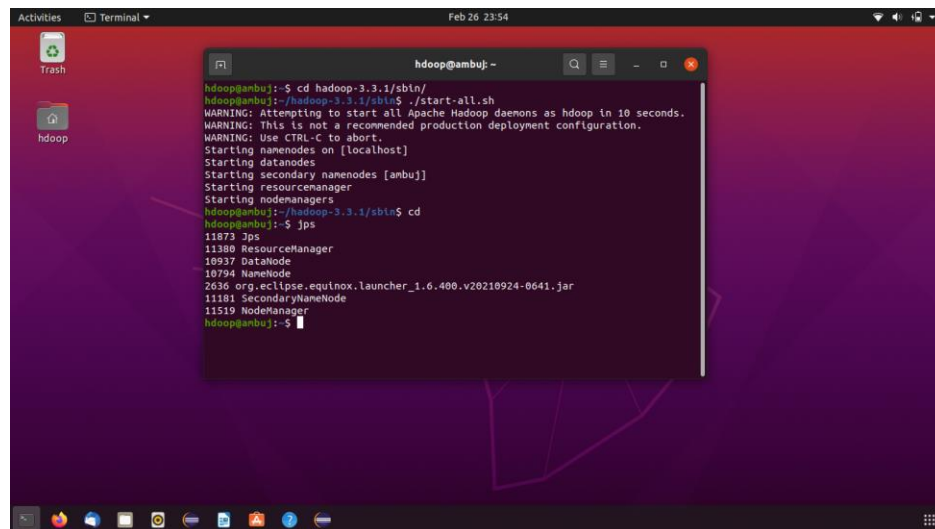
Big Data & Large-Scale Computing

DA-IICT, Gandhinagar

*27-February-2022*

## Environment setup:

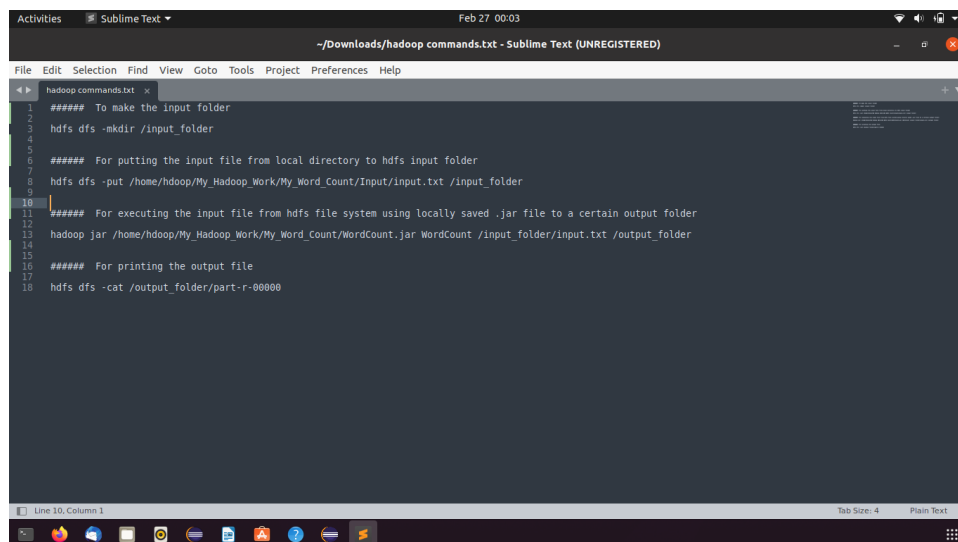
Before moving to the questions, we must first set up the hdfs environment to make all the nodes and resource managers running.



```
hadoop@ambuj:~$ cd hadoop-3.3.1/sbin/
hadoop@ambuj:~/hadoop-3.3.1/sbin$ ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ambuj]
Starting resourcemanager
Starting nodemanagers
hadoop@ambuj:~/hadoop-3.3.1/sbin$ cd
hadoop@ambuj:~$ jps
11873 Jps
11380 ResourceManager
10937 DataNode
10794 NameNode
2636 org.eclipse.equinox.launcher_1.6.400.v20210924-0641.jar
11181 SecondaryNameNode
11519 NodeManager
hadoop@ambuj:~$
```

*Figure 1: Starting the HADOOP environment*

To run the java projects using .jar files on input data in hdfs file system, following commands are used:



```
##### To make the input folder
hdfs dfs -mkdir /input_folder

##### For putting the input file from local directory to hdfs input folder
hdfs dfs -put /home/hadoop/My_Hadoop_Work/My_Word_Count/Input/input.txt /input_folder

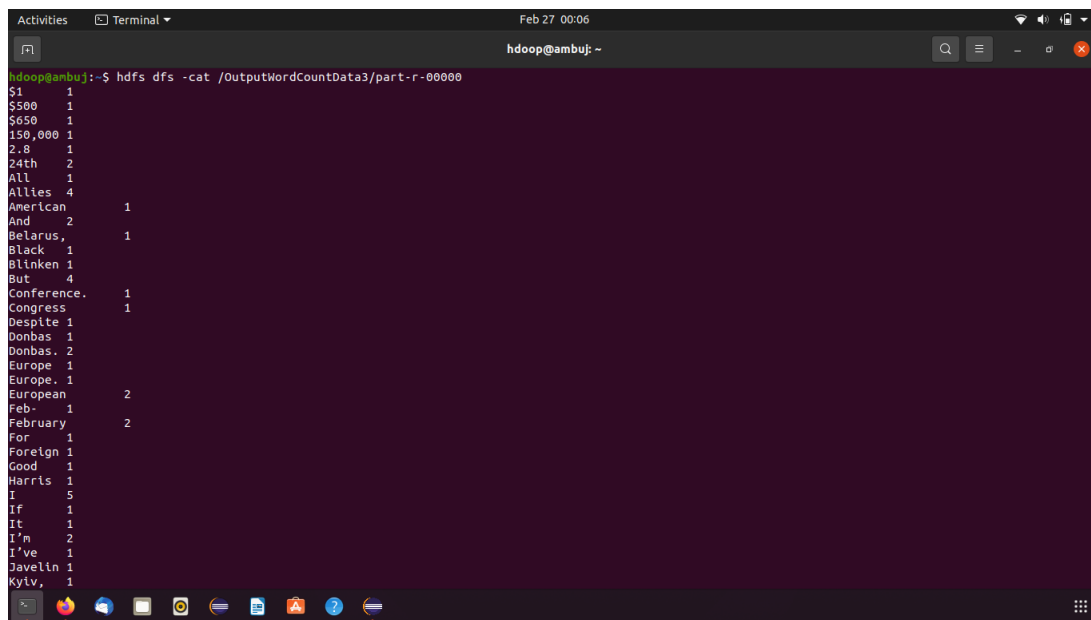
##### For executing the input file from hdfs file system using locally saved .jar file to a certain output folder
hadoop jar /home/hadoop/My_Hadoop_Work/My_Word_Count/WordCount.jar WordCount /input_folder/input.txt /output_folder

##### For printing the output file
hdfs dfs -cat /output_folder/part-r-00000
```

*Figure 2: Commands to run a .jar file in hdfs file system*

## 1<sup>st</sup> Question:

In the 1<sup>st</sup> question, we were asked to modify the WordCount.java file to optimize the results of the out file. We have taken the input.txt file after taking data from the white house press release on Ukraine and Russia issues. We have first run the basic MapReduce WordCount.java file on the input data to check the raw output.



```
hadoop@ambuj:~$ hdfs dfs -cat /OutputWordCountData3/part-r-00000
51 1
5500 1
5650 1
150,000 1
2.8 1
24th 2
All 1
Allies 4
American 1
And 2
Belarus, 1
Black 1
Blinken 1
But 4
Conference. 1
Congress 1
Despite 1
Donbas 1
Donbas. 2
Europe 1
Europe. 1
European 2
Feb. 1
February 2
For 1
Foreign 1
Good 1
Harris 1
I 5
If 1
It 1
I'm 2
I've 1
Javelin 1
Kylv, 1
```

Figure 3: Running non-modified WordCount.java on input.txt

We have made the following changes to the WordCount.java code:

- i. We first changed the method of tokenizing data so that we can also remove the special character symbols from the tokens.

*StringTokenizer itr = new StringTokenizer(line, " \\t\\n\\r\\f,.-:;?![\"]");*

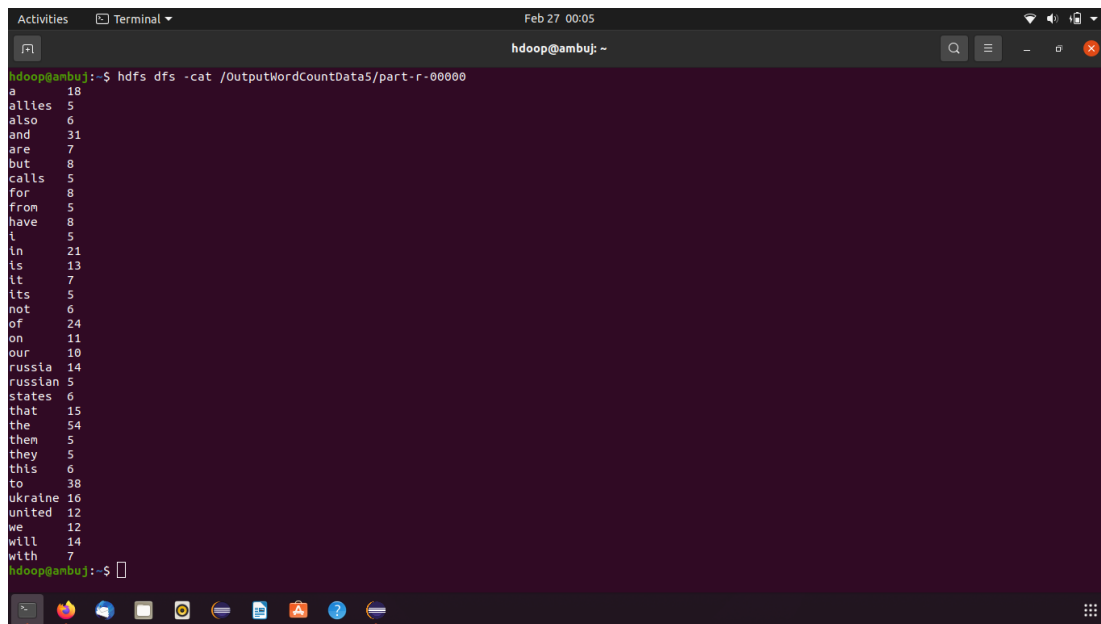
- ii. We do not want the lower-case and upper-case words to be considered distinct words. Therefore, we have used *toLowerCase()* function.

*word.set(itr.nextToken().toLowerCase());*

- iii. We only wanted the words that come more than 4 times in the file therefore, we have put an extra condition where the words are being recorded.

*if (sum > 4) output.collect(key, new IntWritable(sum));*

We have gotten the following results after running the modified WordCount.java file:



The image shows a terminal window with a dark background. The prompt is 'hadoop@anbu: ~'. The command executed is 'hdfs dfs -cat /OutputWordCountData5/part-r-00000'. The output is a list of words and their counts, such as 'a 18', 'allies 5', 'also 6', 'and 31', 'are 7', 'but 8', 'calls 5', 'for 8', 'from 5', 'have 8', 'i 5', 'in 21', 'is 13', 'it 7', 'its 5', 'not 6', 'of 24', 'on 11', 'our 10', 'russia 14', 'russian 5', 'states 6', 'that 15', 'the 54', 'them 5', 'they 5', 'this 6', 'to 38', 'ukraine 16', 'united 12', 'we 12', 'will 14', and 'with 7'. The terminal window has a title bar with 'Activities', 'Terminal', and a date/time 'Feb 27 00:05'. There are also window control buttons and a search icon.

```
hadoop@anbu:~$ hdfs dfs -cat /OutputWordCountData5/part-r-00000
a 18
allies 5
also 6
and 31
are 7
but 8
calls 5
for 8
from 5
have 8
i 5
in 21
is 13
it 7
its 5
not 6
of 24
on 11
our 10
russia 14
russian 5
states 6
that 15
the 54
them 5
they 5
this 6
to 38
ukraine 16
united 12
we 12
will 14
with 7
hadoop@anbu:~$
```

Figure 4: Running modified WordCount.java on input.txt

## 2<sup>nd</sup> Question:

We have used the link <https://www.nber.org/research/data/us-patents> to download the pairwise citation data. The dataset contains the citations and cited files. We are using the modified version of MapReduce WordCount algorithm to calculate the list of citations.

We have used MyJob.java file to calculate the list of citations. Following steps are followed to calculate the results of the list of citations:

- i. We have calculated the list of citations that which files have cited the key file. We can further change the reduce function to change this into count of time the key file is cited.

```

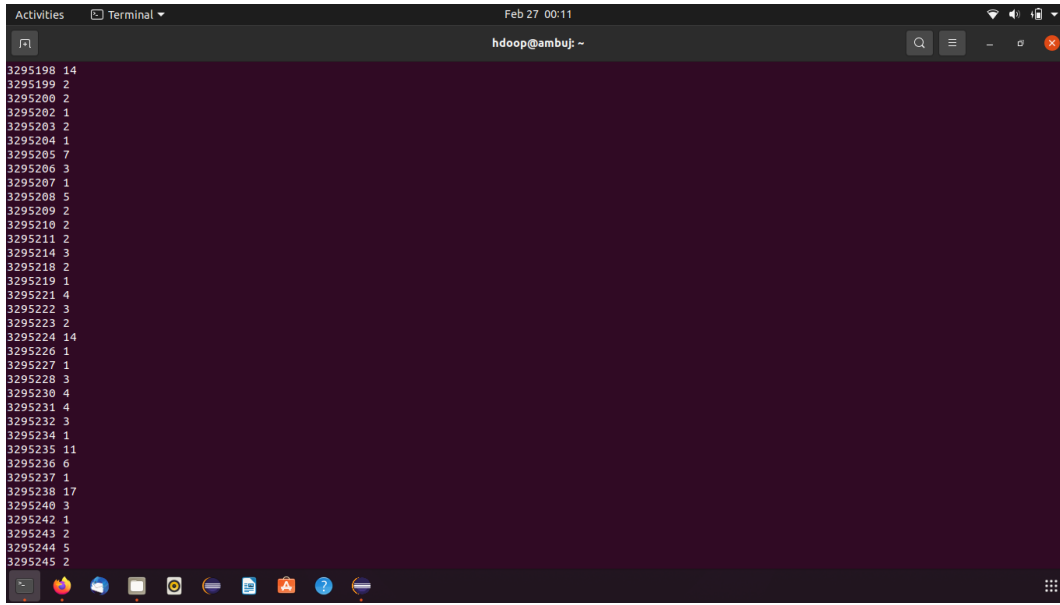
Activities  Terminal  Feb 27 00:09
hadoop@ambuj: ~

3823186 3900567
3823189 4460604
382319 4545422,4577672
3823190 4643903,4585582,4537764,4144200,4163109
3823191 4117017,4001291,5030679,5284886
3823192 4357344,4594360,3950433,4004023,4017640,3864405
3823193 4788351
3823195 4060469,4400550,4716255,5811605
3823196 4159256,4159965,4159963
3823198 4313016,4242530,4012456,4469911
3823199 4536604,4687876,4377719,5100854,4992609,5043504,4229606,4177220,4334117,4229605,4243829
3823200 4145369,3925297,4701481
3823201 4390645,4403908,4683272,4647624,4690956,4542165,4593051,4357430,4504633,4742113,4623674,4477603,4652589,4359542,4585831,4460715,4640935,450067
5,4554295,4431754,4594366,4689354,4190711,4014846,4172825,4327005,3953393,4210727,4358700,4226756,4202956,4264755,4134884,4181781,4242249,4342840,4049
590,4125505,4327194,4198488,4208314,4334049,4104236,4148840,4332716,4153643,4338407,4021383,4176218,4312963,4312973,4165432,4021384,3933937,4282331,49
97857,4797501,5081180,5021507,5854358,5395888,5196476,5192812,5451631,5200434,5280766,5382642,5482995,5512633,5060603,5516844,5521249,5554662,5594066
3823202 4169113,4014828,4014850
3823203 4116917,4039629,4096203,4101482,4163765,4320084,4000341,3970771,4237245,4156673,3865776,4102849,3994993,4560498,4427834,5118762,5149895,491416
0,5705571,5633415,5457161,5637783,5545783,5278252,5223579,5241008,5187236,5780540,6005050
3823204 5278234,4877087,4859727,4605706,4418176,4504659
3823205 4389502,4576977,4361528,4936936,5055346,4173558,4255308,4355071,4279789,4208465,4112023,3998768,4173559,4152189,5147453,5322715
3823206 3998789,3989772
3823207 5235080,5532401
3823208 4233396,3962395,4783293,4397048,4409972
3823209 4025277,5595690,5177340
382321 4678732,4565240
3823210 4704238,4065594,5958322
3823211 5660922,5605717,5589122,5596602,4100237,4069924,4398581,4399886,4935889,4847148
3823212 4798611,4834734,5039414,4416814,4409332,4591456,4291013,4347234,4193813,4264493,4349470,5700476,5660857,5466462,5827840
3823213 4108934,4031179,4246211,4255372,4275023,3953558,4247650,4007246,4126662,4504601,5085814
3823214 4187131,4214028,4187338,4217385,4248922,4082876
3823215 3884605,3923438,4041119,4141947,4196163,4094946,3976736,4769193,4459094,4474545
3823216 4693483,4637618,4723905,4625383,4395379,5099888,4803033,4826028,3923952,4102623,3986810,4170448,4247277,4107249,3929538,4336959,4120521,423198
3,5756023,5879723,5637332,5362107,5286064
3823217 5185594,5196145,5313185,5800668,5800768,5802709,5849137,5849129,5864280,5993990,5554679,4055526,4348584,4334148,4277673,4074222,4330703,432748
0,4242573,4318220,4318881,4286376,4177376,4017715,4889975,5057673,5025131,5093898,4764664,4857880,4954695,4876440,4935156,4866253,4543474,4693940,4444

```

Figure 5: Non-modified MyJob.java returning list of citations

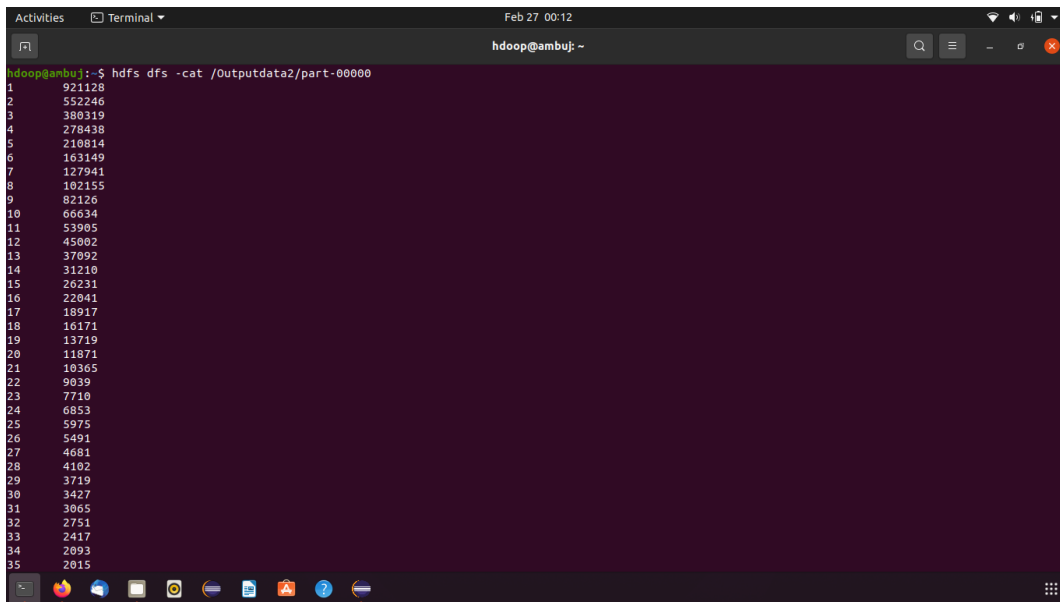
- ii. We changed the reduce function to convert the list into the count of times this key file is cited.



```
hadoop@ambuj: ~$ cat /outputdata2/part-00000
3295198 14
3295199 2
3295200 2
3295202 1
3295203 2
3295204 1
3295205 7
3295206 3
3295207 1
3295208 5
3295209 2
3295210 2
3295211 2
3295214 3
3295218 2
3295219 1
3295221 4
3295222 3
3295223 2
3295224 14
3295226 1
3295227 1
3295228 3
3295230 4
3295231 4
3295232 3
3295234 1
3295235 11
3295236 6
3295237 1
3295238 17
3295240 3
3295242 1
3295243 2
3295244 5
3295245 2
```

Figure 6: Modified MyJob.java returning count of citations

- iii. We have further used CitationHistogram.java file to create histogram data based on the output file that is generated after *part ii*. The output file after this step contains the number of files that have been cited once. Similarly, the number of files that have been cited twice, thrice, and so on.



```
hadoop@ambuj: ~$ hdfs dfs -cat /Outputdata2/part-00000
1 921128
2 552246
3 380319
4 278438
5 210814
6 163149
7 127941
8 102155
9 82126
10 66634
11 53905
12 45002
13 37092
14 31210
15 26231
16 22041
17 18917
18 16171
19 13719
20 11871
21 10365
22 9939
23 7710
24 6853
25 5975
26 5491
27 4681
28 4102
29 3719
30 3427
31 3865
32 2751
33 2417
34 2093
35 2015
```

Figure 7: CitationHistogram.java file to create histogram data

We can also use this data to plot histogram.

---