LOG8415
Advanced Concepts of Cloud Computing
Lab2: MapReduce with Hadoop and Spark on Microsoft Azure

Marwane Adala
Vahid Majdinasab
Chun-An Bau
{marwane.adala, vahid.majdinasab, chun-an.bau}@polymtl.ca

We introduced how to set up the system, including Hadoop and Spark in section 1, and recorded what should be noticed while performing the WordCount experimentations in different comparison scenarios in sections 2 and 3. In section 4, we described how we managed to solve the social problem to implement a simple "People You Might Know" social network friendship recommendation algorithm. Finally, we showed the procedures of running all the scenarios and benchmarks in section 5. The setup procedures and corresponding scripts could be found in our GitHub repository [4].

1 Environment Setup

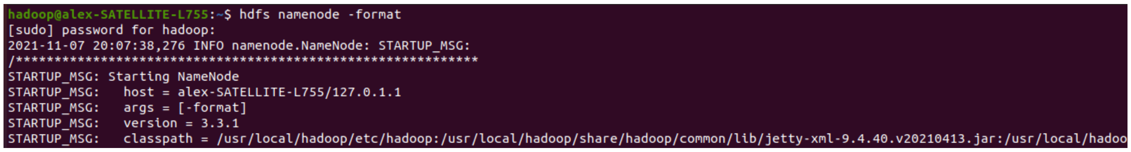
We set up our testing system environment(s) in the order of **Hadoop**, and **Spark**.

1.1 Hadoop Setup [5], [8], [6], [7] and [1]

We used a 20.04 LTS Ubuntu OS (local machine) with Openjdk version "1.8.0_292" and OpenSSH server and client installed on it. Our goal was to configure Hadoop on our system using standalone mode. The first steps were to create a Hadoop user, and add him to the sudo group. Then, we should configure a Password-less SSH using this user.
See the following commands concerning password-less SSH :

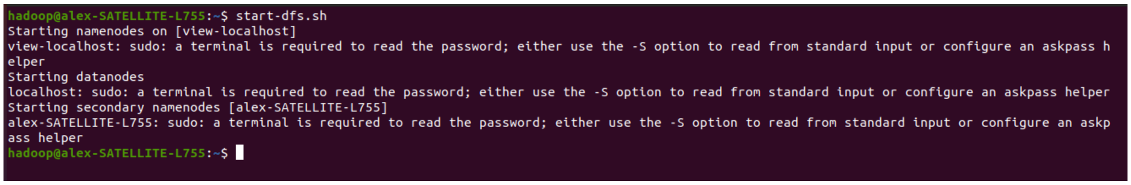
```
$ ssh-keygen -t rsa
$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ sudo chmod 640 ~/.ssh/authorized_keys
$ ssh localhost #to verify if the password-less SSH is functional
```

After downloading, extracting and dealing with permissions of Hadoop version 3.3.1, we made the essential configurations concerning Hadoop and Java environment variables. We also edited core-site.xml, hdfs-site.xml, mapred-site.xml, and yarn-site.xml XML files to specify NameNode URL, define the storing location for node metadata and fsimage files, define MapReduce values and YARN-related settings. See Figures 1, 2, 3, and 4 for the validation of these steps. We refer the reader to Appendix A to see the other aspects of Hadoop configuration in standalone mode.



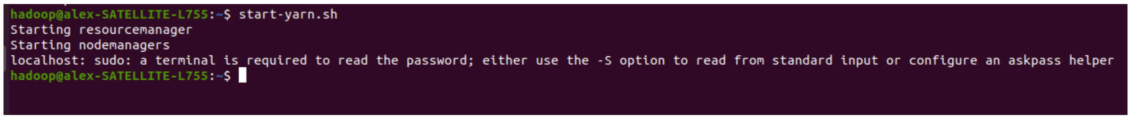
```
hadoop@alex-SATELLITE-L755:~$ hdfs namenode -format
[sudo] password for hadoop:
2021-11-07 20:07:38,276 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = alex-SATELLITE-L755/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.1
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/jetty-xml-9.4.40.v20210413.jar:/usr/local/hadoo
```

Figure 1: Hadoop Configuration Validation (1)



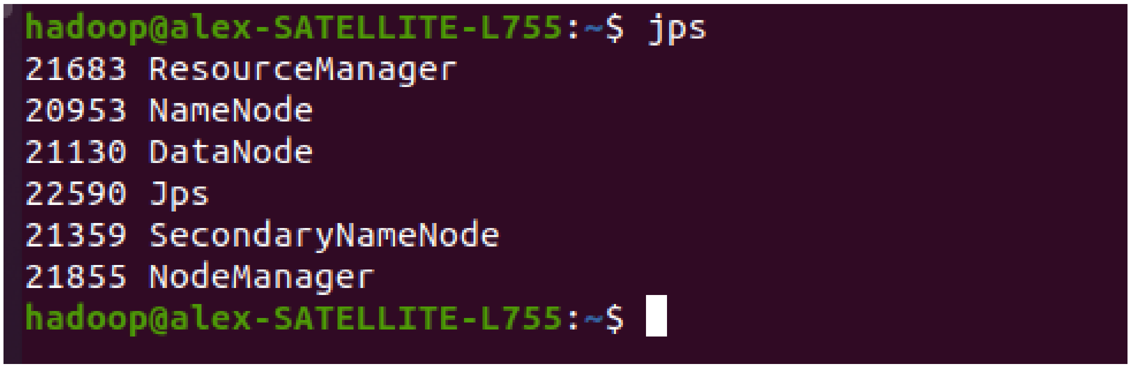
```
hadoop@alex-SATELLITE-L755:~$ start-dfs.sh
Starting namenodes on [view-localhost]
view-localhost: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass h
elper
Starting datanodes
localhost: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
Starting secondary namenodes [alex-SATELLITE-L755]
alex-SATELLITE-L755: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askp
ass helper
hadoop@alex-SATELLITE-L755:~$ 
```

Figure 2: Hadoop Configuration Validation (2)



```
hadoop@alex-SATELLITE-L755:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
localhost: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
hadoop@alex-SATELLITE-L755:~$ 
```

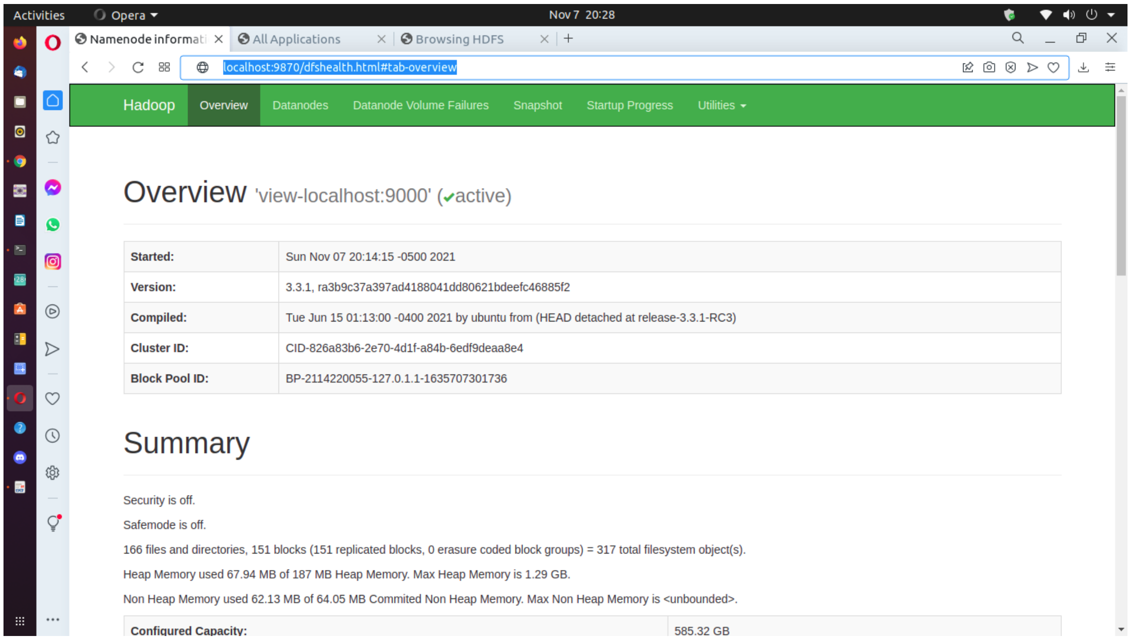
Figure 3: Hadoop Configuration Validation (3)



```
hadoop@alex-SATELLITE-L755:~$ jps
21683 ResourceManager
20953 NameNode
21130 DataNode
22590 Jps
21359 SecondaryNameNode
21855 NodeManager
hadoop@alex-SATELLITE-L755:~$ 
```

Figure 4: Hadoop Configuration Validation(4)

In Figures 5, 6 and 7, we can observe that we can access Hadoop Namenode (using port 9870), the Resource Manage (using port 8088) and browsing HDFS filesystem.



Overview 'view-localhost:9000' (✓active)

Started:	Sun Nov 07 20:14:15 -0500 2021
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 01:13:00 -0400 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-826a83b6-2e70-4d1f-a84b-6edf9deaa8e4
Block Pool ID:	BP-2114220055-127.0.1.1-1635707301736

Summary

Security is off.

Safemode is off.

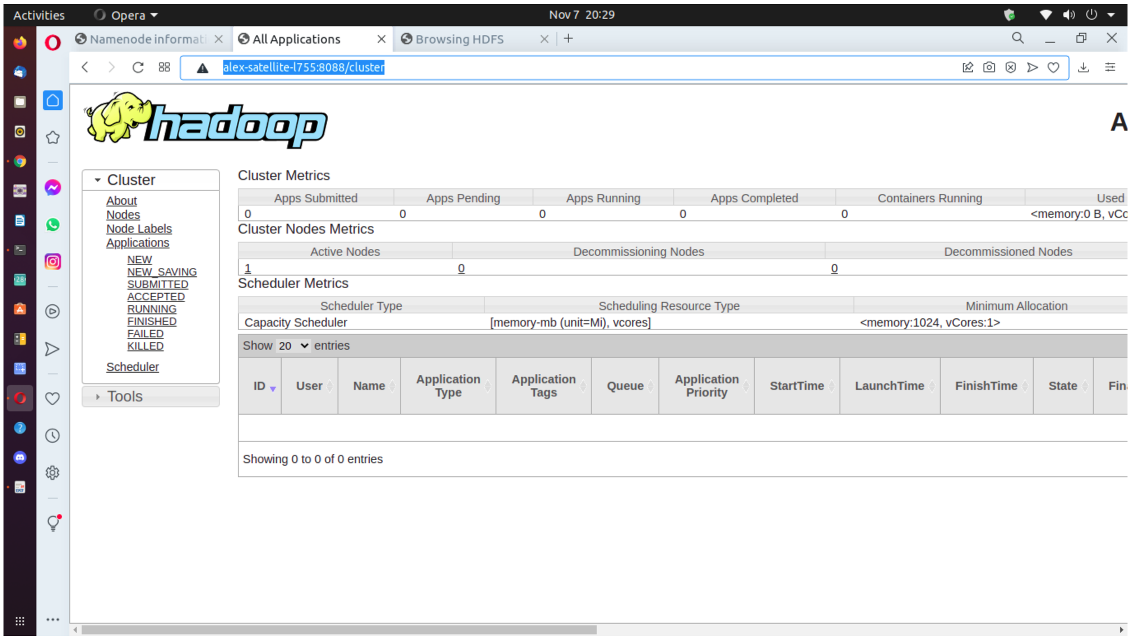
166 files and directories, 151 blocks (151 replicated blocks, 0 erasure coded block groups) = 317 total filesystem object(s).

Heap Memory used 67.94 MB of 187 MB Heap Memory. Max Heap Memory is 1.29 GB.

Non Heap Memory used 62.13 MB of 64.05 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	585.32 GB

Figure 5: Hadoop Namenode



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used
0	0	0	0	0	<memory:0 B, vCo

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

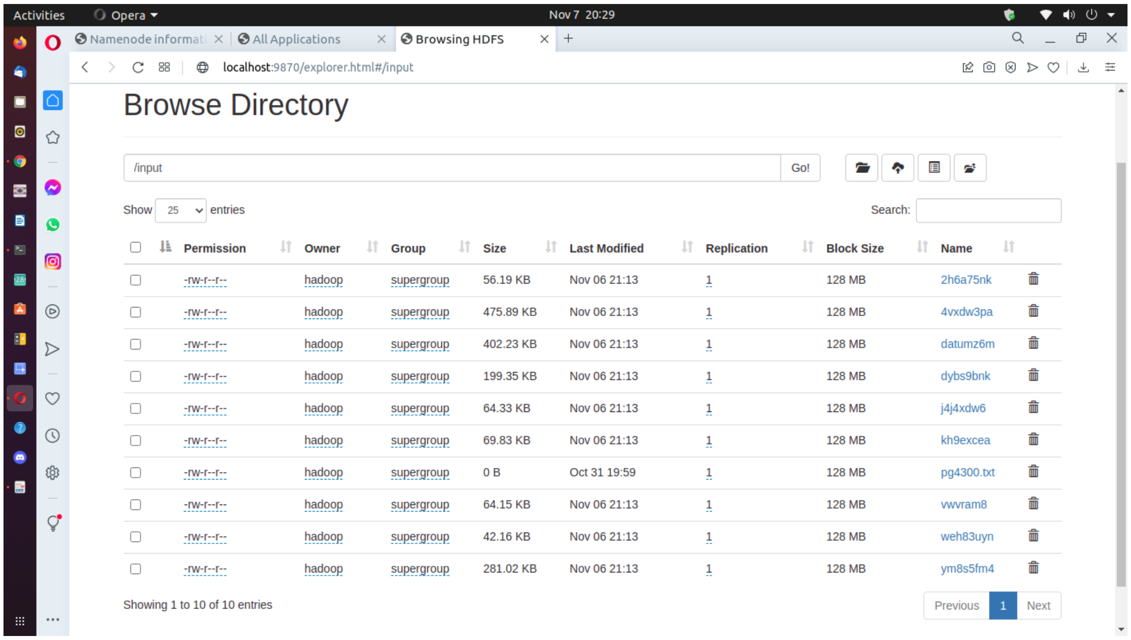
Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	Fin

Showing 0 to 0 of 0 entries

Figure 6: Resource Manager



Browse Directory

/input

Show 25 entries   Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
☐	-rw-r--r--	hadoop	supergroup	56.19 KB	Nov 06 21:13	1	128 MB	2h6a75nk	🗑
☐	-rw-r--r--	hadoop	supergroup	475.89 KB	Nov 06 21:13	1	128 MB	4vxdw3pa	🗑
☐	-rw-r--r--	hadoop	supergroup	402.23 KB	Nov 06 21:13	1	128 MB	datumz6m	🗑
☐	-rw-r--r--	hadoop	supergroup	199.35 KB	Nov 06 21:13	1	128 MB	dybs9bnk	🗑
☐	-rw-r--r--	hadoop	supergroup	64.33 KB	Nov 06 21:13	1	128 MB	j4j4xdw6	🗑
☐	-rw-r--r--	hadoop	supergroup	69.83 KB	Nov 06 21:13	1	128 MB	kh9excea	🗑
☐	-rw-r--r--	hadoop	supergroup	0 B	Oct 31 19:59	1	128 MB	pg4300.txt	🗑
☐	-rw-r--r--	hadoop	supergroup	64.15 KB	Nov 06 21:13	1	128 MB	vwvram8	🗑
☐	-rw-r--r--	hadoop	supergroup	42.16 KB	Nov 06 21:13	1	128 MB	weh83uyn	🗑
☐	-rw-r--r--	hadoop	supergroup	281.02 KB	Nov 06 21:13	1	128 MB	ym8s5fm4	🗑

Showing 1 to 10 of 10 entries

Figure 7: Browsing HDFS Filesystem

1.2 Hadoop and Spark Setup on Azure

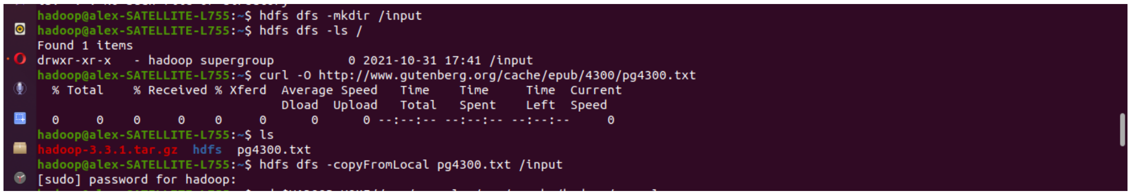
Since only one Hadoop node was required, we did not need additional setting related to network traffic. We could run Hadoop by the instructions in section 1.1. We configured Spark on Azure following the online tutorials [10] [9], and the most essential part is shown as below.

```
conf = SparkConf()
sc = SparkContext(conf=conf)
distFile = sc.textFile(input_file)
count = distFile.flatMap(lambda line: line.split("␣"))   \
        .map(lambda word: (word, 1))    \
        .reduceByKey(lambda a, b: a + b)    \
        .sortByKey()
count.saveAsTextFile(output_file)
```

The *input_file* and *output_file* in the python script above could be either local storage or HDFS. In our experiments, we retrieved the data sets from HDFS and wrote the result back to HDFS.

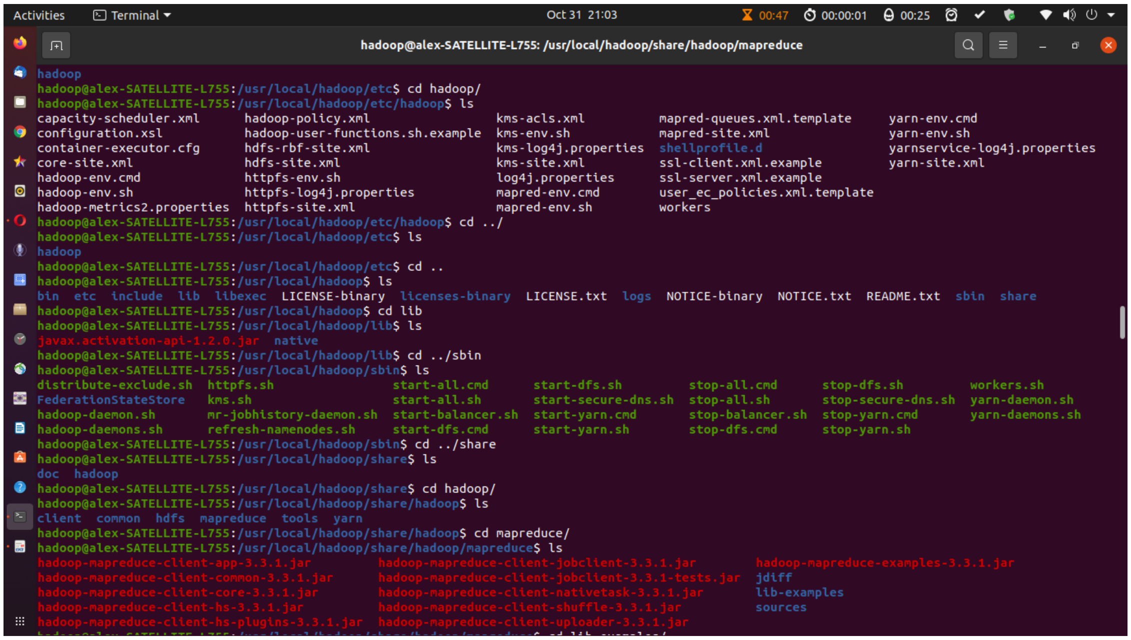
2 Initial Experiments with WordCount program

In this section, we will go over the various results obtained by running some preliminary tests with Hadoop. After creating an input directory within HDFS, we begin by copying James Joyce's Ulysses book (named here pg4300.txt) to HDFS (see Figures 8, 9 and 10). Later on, we run the WordCount program on this book file in Figures 11, 12, 13 and 14. The Resource Manager interface in Figures 15 and 16 allows us to validate the success of these steps.



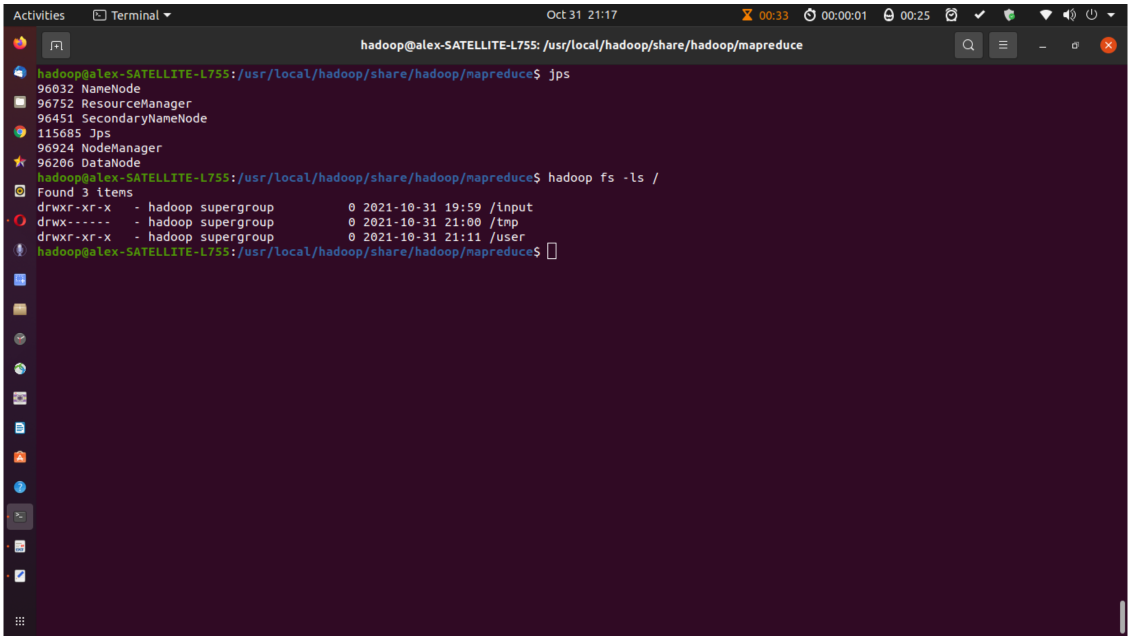
```
hadoop@alex-SATELLITE-L755:~$ hdfs dfs -mkdir /input
hadoop@alex-SATELLITE-L755:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2021-10-31 17:41 /input
hadoop@alex-SATELLITE-L755:~$ curl -O http://www.gutenberg.org/cache/epub/4300/pg4300.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
hadoop@alex-SATELLITE-L755:~$ ls
hadoop-3.3.1.tar.gz  hdfs  pg4300.txt
hadoop@alex-SATELLITE-L755:~$ hdfs dfs -copyFromLocal pg4300.txt /input
[sudo] password for hadoop:
```

Figure 8: Making Input Directory in HDFS and Downloading James Joyce’s Ulysses book


Activities
Terminal
Oct 31 21:03
00:47
00:00:01
00:25
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

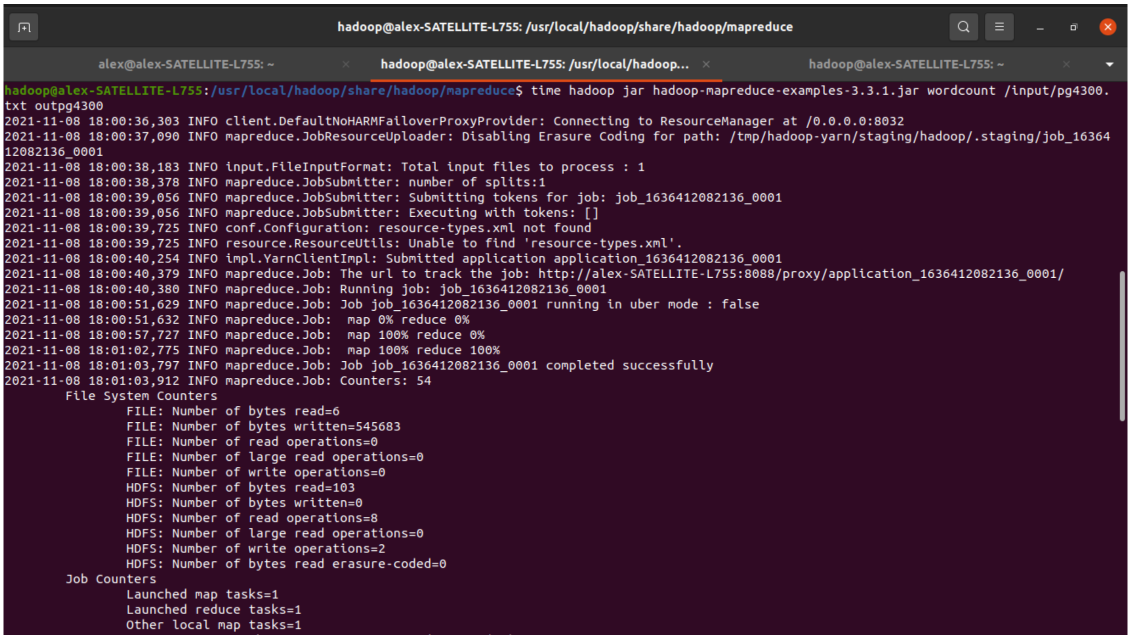
```
hadoop
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/etc$ cd hadoop/
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/etc/hadoop$ ls
capacity-scheduler.xml      hadoop-policy.xml                 kms-acls.xml          mapred-queues.xml.template     yarn-env.cmd
configuration.xsl           hadoop-user-functions.sh.example  kms-env.sh            mapred-site.xml                yarn-env.sh
container-executor.cfg      hdfs-rbf-site.xml                 kms-log4j.properties  shellprofile.d                 yarnservice-log4j.properties
core-site.xml               hdfs-site.xml                     kms-site.xml          ssl-client.xml.example         yarn-site.xml
hadoop-env.cmd              httpfs-env.sh                     log4j.properties      ssl-server.xml.example
hadoop-env.sh               httpfs-log4j.properties           mapred-env.cmd        user_ec_policies.xml.template
hadoop-metrics2.properties  httpfs-site.xml                   mapred-env.sh         workers
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/etc/hadoop$ cd ../
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/etc$ ls
hadoop
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/etc$ cd ..
hadoop@alex-SATELLITE-L755:/usr/local/hadoop$ ls
bin  etc  include  lib  libexec  LICENSE-binary  licenses-binary  LICENSE.txt  logs  NOTICE-binary  NOTICE.txt  README.txt  sbin  share
hadoop@alex-SATELLITE-L755:/usr/local/hadoop$ cd lib
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ ls
javax.activation-api-1.2.0.jar  native
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ cd ../sbin
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/sbin$ ls
distribute-exclude.sh  httpfs.sh                start-all.cmd      start-dfs.sh         stop-all.cmd      stop-dfs.sh         workers.sh
FederationStateStore   kms.sh                   start-all.sh       start-secure-dns.sh  stop-all.sh       stop-secure-dns.sh  yarn-daemon.sh
hadoop-daemon.sh       mr-jobhistory-daemon.sh  start-balancer.sh  start-yarn.cmd       stop-balancer.sh  stop-yarn.cmd       yarn-daemons.sh
hadoop-daemons.sh      refresh-namenodes.sh     start-dfs.cmd      start-yarn.sh        stop-dfs.cmd      stop-yarn.sh
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/sbin$ cd ../share
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share$ ls
doc  hadoop
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share$ cd hadoop/
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop$ ls
client  common  hdfs  mapreduce  tools  yarn
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop$ cd mapreduce/
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ ls
hadoop-mapreduce-client-app-3.3.1.jar         hadoop-mapreduce-client-jobclient-3.3.1.jar        hadoop-mapreduce-examples-3.3.1.jar
hadoop-mapreduce-client-common-3.3.1.jar      hadoop-mapreduce-client-jobclient-3.3.1-tests.jar  jdiff
hadoop-mapreduce-client-core-3.3.1.jar        hadoop-mapreduce-client-nativetask-3.3.1.jar       lib-examples
hadoop-mapreduce-client-hs-3.3.1.jar          hadoop-mapreduce-client-shuffle-3.3.1.jar          sources
hadoop-mapreduce-client-hs-plugins-3.3.1.jar  hadoop-mapreduce-client-uploader-3.3.1.jar
```

Figure 9: Exploring Hadoop Related Directories


Activities
Terminal
Oct 31 21:17
00:33
00:00:01
00:25
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

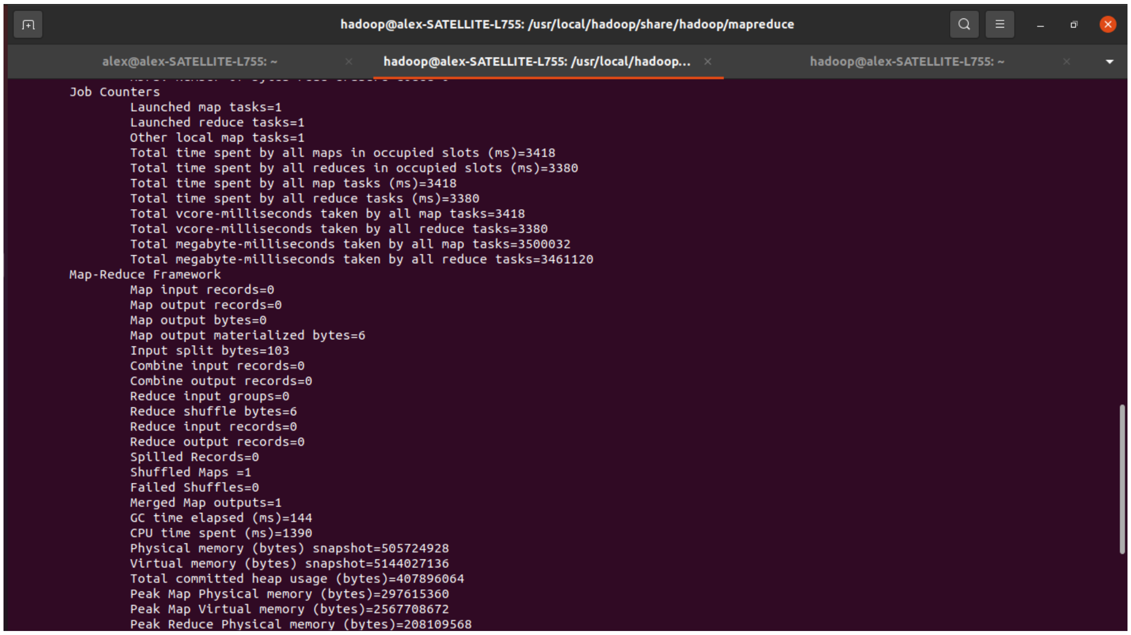
```
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ jps
96032 NameNode
96752 ResourceManager
96451 SecondaryNameNode
115685 Jps
96924 NodeManager
96206 DataNode
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ hadoop fs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2021-10-31 19:59 /input
drwx------   - hadoop supergroup          0 2021-10-31 21:00 /tmp
drwxr-xr-x   - hadoop supergroup          0 2021-10-31 21:11 /user
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ 
```

Figure 10: Check out all Hadoop Daemons And Exploring HDFS files


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
alex@alex-SATELLITE-L755: ~
hadoop@alex-SATELLITE-L755: /usr/local/hadoop...
hadoop@alex-SATELLITE-L755: ~

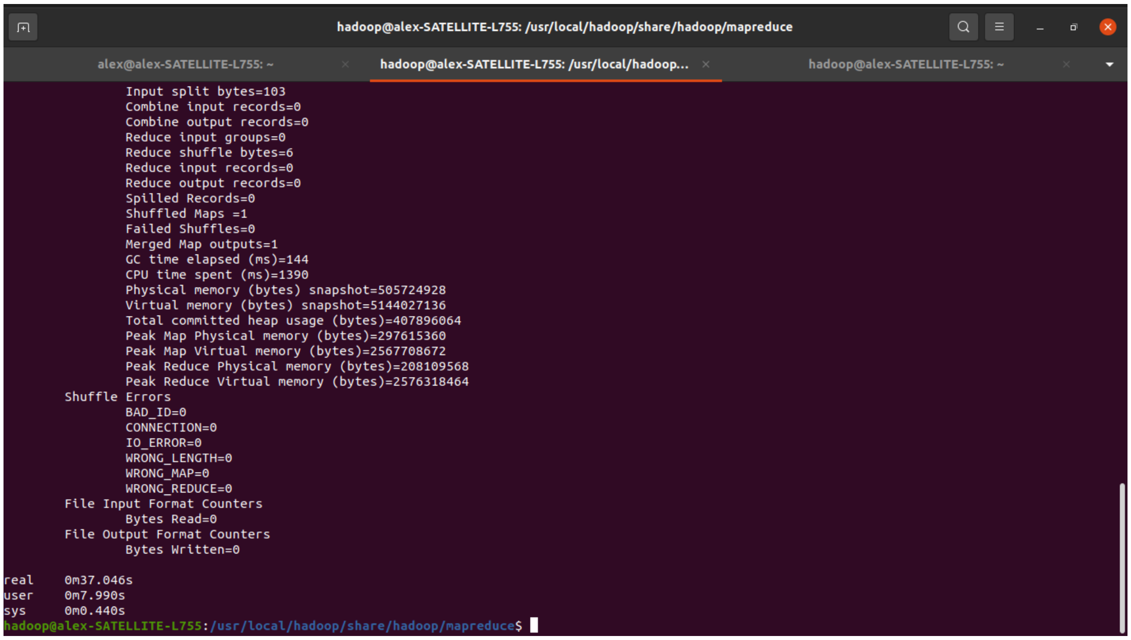
```
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ time hadoop jar hadoop-mapreduce-examples-3.3.1.jar wordcount /input/pg4300.
txt outpg4300
2021-11-08 18:00:36,303 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-08 18:00:37,090 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_16364
12082136_0001
2021-11-08 18:00:38,183 INFO input.FileInputFormat: Total input files to process : 1
2021-11-08 18:00:38,378 INFO mapreduce.JobSubmitter: number of splits:1
2021-11-08 18:00:39,056 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1636412082136_0001
2021-11-08 18:00:39,056 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-08 18:00:39,725 INFO conf.Configuration: resource-types.xml not found
2021-11-08 18:00:39,725 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-11-08 18:00:40,254 INFO impl.YarnClientImpl: Submitted application application_1636412082136_0001
2021-11-08 18:00:40,379 INFO mapreduce.Job: The url to track the job: http://alex-SATELLITE-L755:8088/proxy/application_1636412082136_0001/
2021-11-08 18:00:40,380 INFO mapreduce.Job: Running job: job_1636412082136_0001
2021-11-08 18:00:51,629 INFO mapreduce.Job: Job job_1636412082136_0001 running in uber mode : false
2021-11-08 18:00:51,632 INFO mapreduce.Job:  map 0% reduce 0%
2021-11-08 18:00:57,727 INFO mapreduce.Job:  map 100% reduce 0%
2021-11-08 18:01:02,775 INFO mapreduce.Job:  map 100% reduce 100%
2021-11-08 18:01:03,797 INFO mapreduce.Job: Job job_1636412082136_0001 completed successfully
2021-11-08 18:01:03,912 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=6
                FILE: Number of bytes written=545683
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=103
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Other local map tasks=1
```

Figure 11: Performing WordCount Java Program on James Joyce's Ulysses book


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
alex@alex-SATELLITE-L755: ~
hadoop@alex-SATELLITE-L755: /usr/local/hadoop...
hadoop@alex-SATELLITE-L755: ~

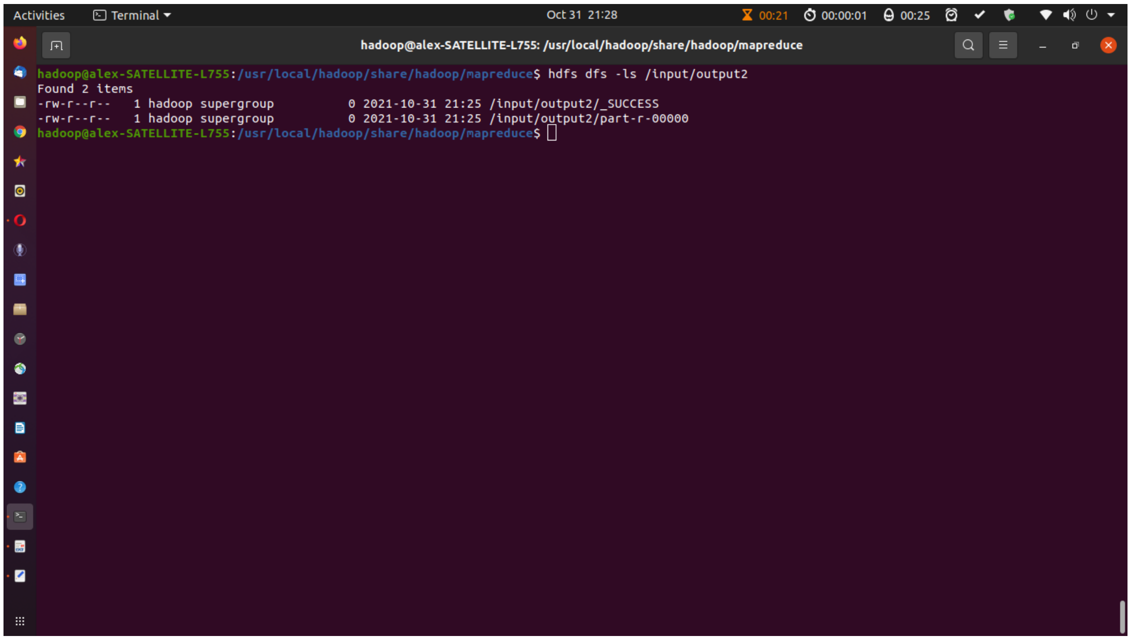
```
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Other local map tasks=1
		Total time spent by all maps in occupied slots (ms)=3418
		Total time spent by all reduces in occupied slots (ms)=3380
		Total time spent by all map tasks (ms)=3418
		Total time spent by all reduce tasks (ms)=3380
		Total vcore-milliseconds taken by all map tasks=3418
		Total vcore-milliseconds taken by all reduce tasks=3380
		Total megabyte-milliseconds taken by all map tasks=3500032
		Total megabyte-milliseconds taken by all reduce tasks=3461120
	Map-Reduce Framework
		Map input records=0
		Map output records=0
		Map output bytes=0
		Map output materialized bytes=6
		Input split bytes=103
		Combine input records=0
		Combine output records=0
		Reduce input groups=0
		Reduce shuffle bytes=6
		Reduce input records=0
		Reduce output records=0
		Spilled Records=0
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		GC time elapsed (ms)=144
		CPU time spent (ms)=1390
		Physical memory (bytes) snapshot=505724928
		Virtual memory (bytes) snapshot=5144027136
		Total committed heap usage (bytes)=407896064
		Peak Map Physical memory (bytes)=297615360
		Peak Map Virtual memory (bytes)=2567708672
		Peak Reduce Physical memory (bytes)=208109568
```

Figure 12: Performing WordCount Java Program on James Joyce's Ulysses book (2)


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
alex@alex-SATELLITE-L755: ~
hadoop@alex-SATELLITE-L755: /usr/local/hadoop...
hadoop@alex-SATELLITE-L755: ~

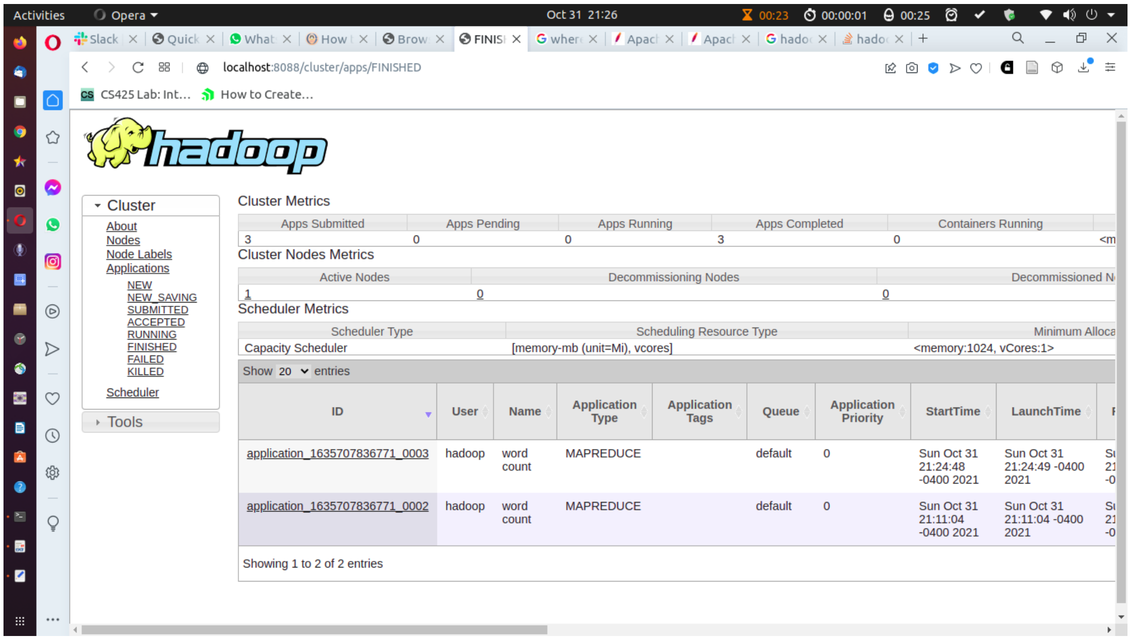
```
		Input split bytes=103
		Combine input records=0
		Combine output records=0
		Reduce input groups=0
		Reduce shuffle bytes=6
		Reduce input records=0
		Reduce output records=0
		Spilled Records=0
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		GC time elapsed (ms)=144
		CPU time spent (ms)=1390
		Physical memory (bytes) snapshot=505724928
		Virtual memory (bytes) snapshot=5144027136
		Total committed heap usage (bytes)=407896064
		Peak Map Physical memory (bytes)=297615360
		Peak Map Virtual memory (bytes)=2567708672
		Peak Reduce Physical memory (bytes)=208109568
		Peak Reduce Virtual memory (bytes)=2576318464
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=0
	File Output Format Counters
		Bytes Written=0

real	0m37.046s
user	0m7.990s
sys	0m0.440s
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ 
```

Figure 13: Performing WordCount Java Program on James Joyce's Ulysses book (3)


Activities
Terminal
Oct 31 21:28
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

```
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ hdfs dfs -ls /input/output2
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2021-10-31 21:25 /input/output2/_SUCCESS
-rw-r--r--   1 hadoop supergroup          0 2021-10-31 21:25 /input/output2/part-r-00000
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$
```

Figure 14: Checking the Output Directory Generated after running WordCount Program



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	
3	0	0	3	0	<m

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned N
1	0	0

Scheduler Metrics

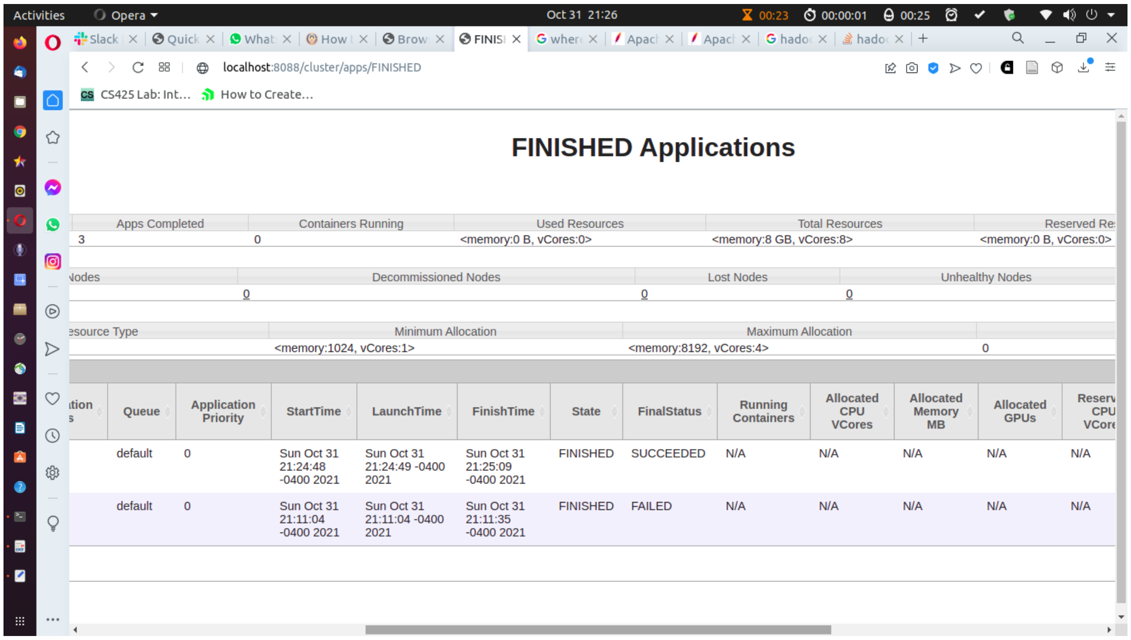
Scheduler Type	Scheduling Resource Type	Minimum Alloca
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	F
application_1635707836771_0003	hadoop	word count	MAPREDUCE		default	0	Sun Oct 31 21:24:48 -0400 2021	Sun Oct 31 21:24:49 -0400 2021	Su 21 -0
application_1635707836771_0002	hadoop	word count	MAPREDUCE		default	0	Sun Oct 31 21:11:04 -0400 2021	Sun Oct 31 21:11:04 -0400 2021	Su 21 -0

Showing 1 to 2 of 2 entries

Figure 15: Checking Resource Manager Jobs (1)



FINISHED Applications

Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Re
3	0	<memory:0 B, vCores:0>	<memory:8 GB, vCores:8>	<memory:0 B, vCores:0>

Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
	0	0	0

esource Type	Minimum Allocation	Maximum Allocation	
	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

ation s	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserv CPU VCore
	default	0	Sun Oct 31 21:24:48 -0400 2021	Sun Oct 31 21:24:49 -0400 2021	Sun Oct 31 21:25:09 -0400 2021	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A
	default	0	Sun Oct 31 21:11:04 -0400 2021	Sun Oct 31 21:11:04 -0400 2021	Sun Oct 31 21:11:35 -0400 2021	FINISHED	FAILED	N/A	N/A	N/A	N/A	N/A

Figure 16: Checking Resource Manager Jobs (2)

3 Further Experiments with WordCount program

3.1 Performance comparison of Hadoop vs. Linux

We ran WordCount Java program example to perform the required tests using these commands, which enabled us to perform 3 experiments on all datasets and measure the user time at each experiment. The Table 1, comparing Linux vs Hadoop, shows the average user time for the nine given datasets. Using the same table and refering to [2] and [3], it is possible to conclude that Linux vastly outperforms Hadoop. In fact, several linux command line tools, such as grep, awk, sort, and uniq, can perform stream processing, so no batching is required and the memory overhead is minimal. Thus, it can be a very simple and rapid approach to pre-process big volumes of data on a local system. Hadoop, on the other hand, is designed to run on lots of machines and process large datasets that would otherwise be impossible to process on a single system.

Table 1: Execution time: Hadoop vs. Linux (sec)

	Hadoop	Linux
Dataset1	8.257	0.254
Dataset2	8.011	0.041
Dataset3	8.070	0.088
Dataset4	8.234	0.213
Dataset5	8.585	0.041
Dataset6	8.161	0.15
Dataset7	8.063	0.04
Dataset8	8.033	0.033
Dataset9	8.437	0.035

The reader is directed to Figures 17 and 18 for more detailed visualizations of the comparison results.The reader should also refer to appendix B for the commands and steps used in this scenario.

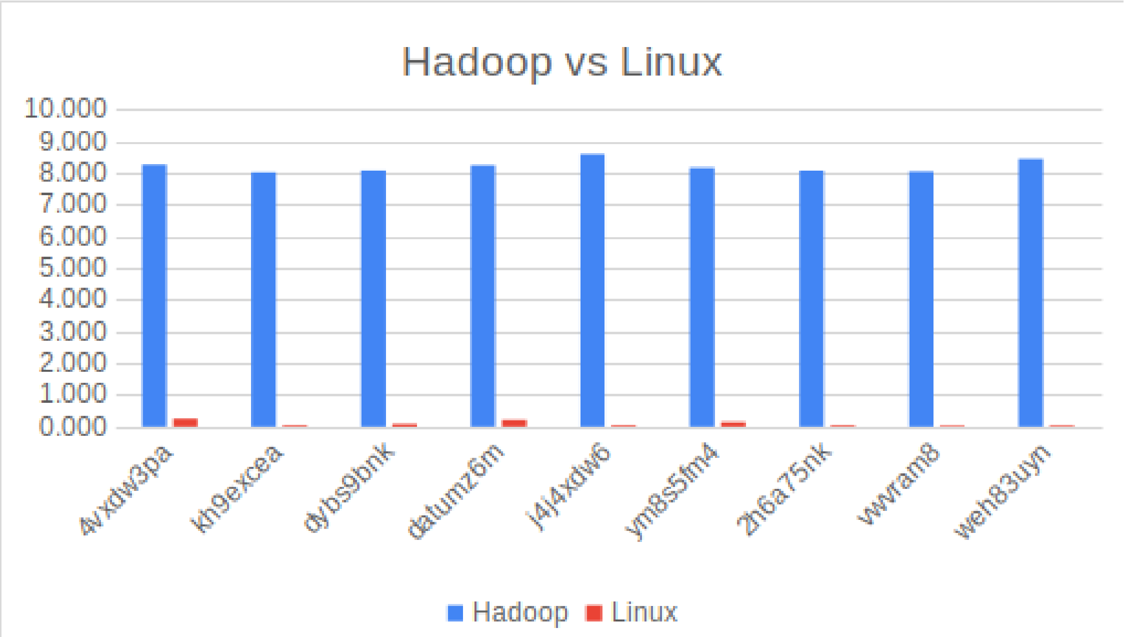


Figure 17: Hadoop vs Linux Comparison. User Time in seconds (1)

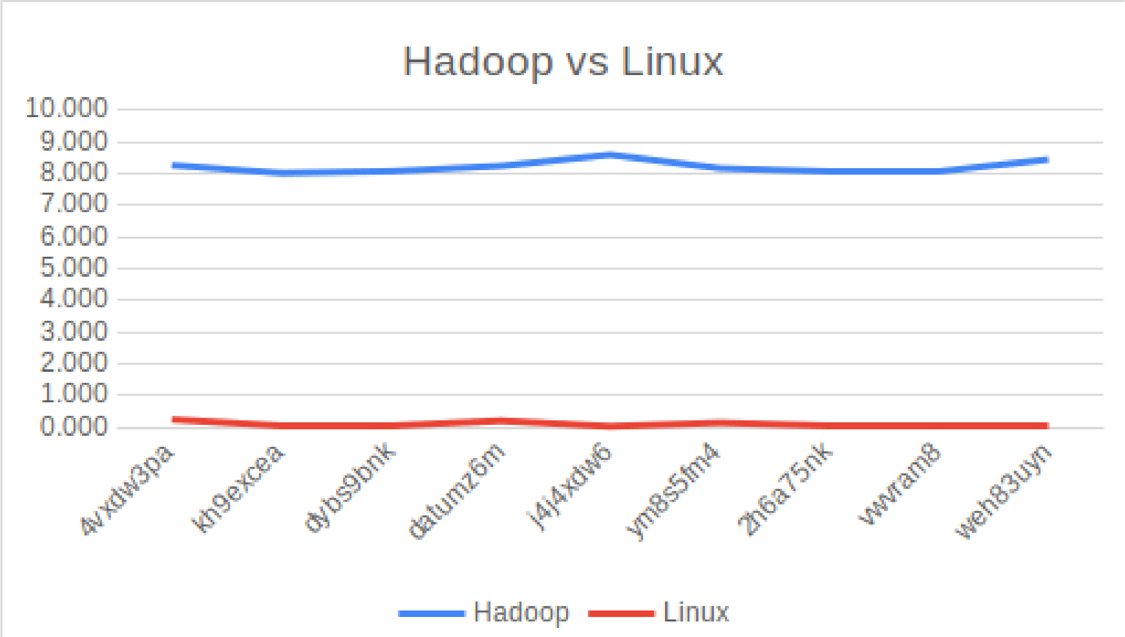


Figure 18: Hadoop vs Linux Comparison. User Time in seconds (2)

3.2 Performance comparison of Hadoop vs. Spark on Azure

As shown in Table 2, the execution time of Hadoop is significantly longer than Spark's, which matches our expectations. Figures 19 and 20 provide more detailed visualizations of the comparison results. Although both retrieved data from HDFS, Spark stored temporary results in RAM while Hadoop wrote back to HDFS during processing stages, which might be the primary reason for the performance difference.

Table 2: Execution time: Hadoop vs. Spark (sec)

	Hadoop	Spark
Dataset 1	11.007	0.154
Dataset 2	9.781	0.153
Dataset 3	10.495	0.159
Dataset 4	10.701	0.143
Dataset 5	10.761	0.164
Dataset 6	10.937	0.158
Dataset 7	10.579	0.157
Dataset 8	10.513	0.158
Dataset 9	10.208	0.149

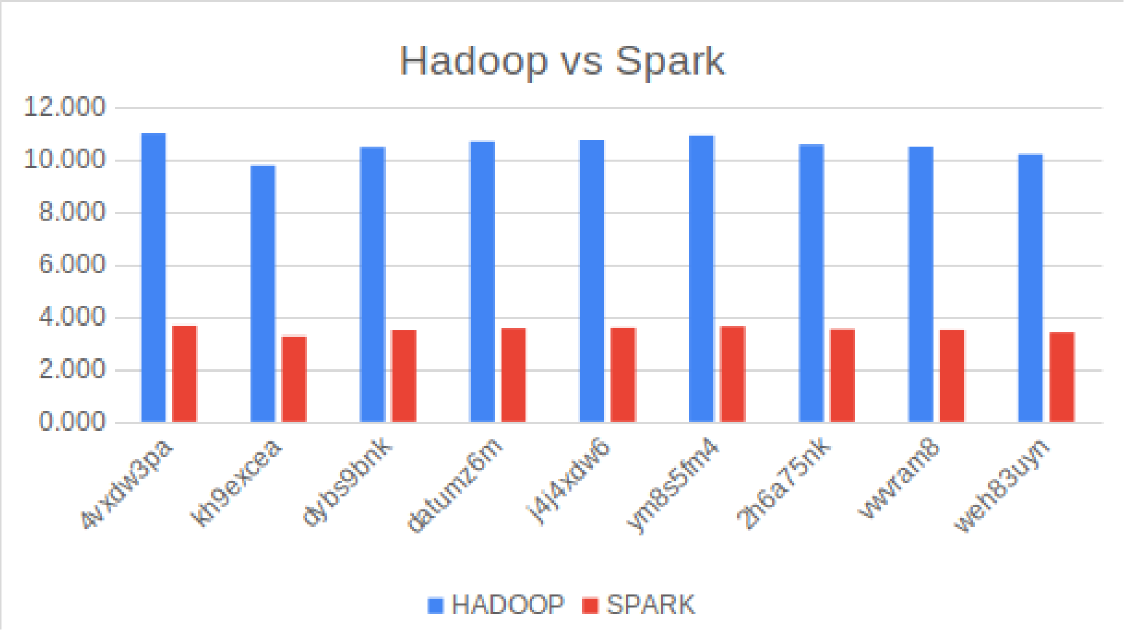


Figure 19: Hadoop vs Spark Comparison. User Time in seconds (1)

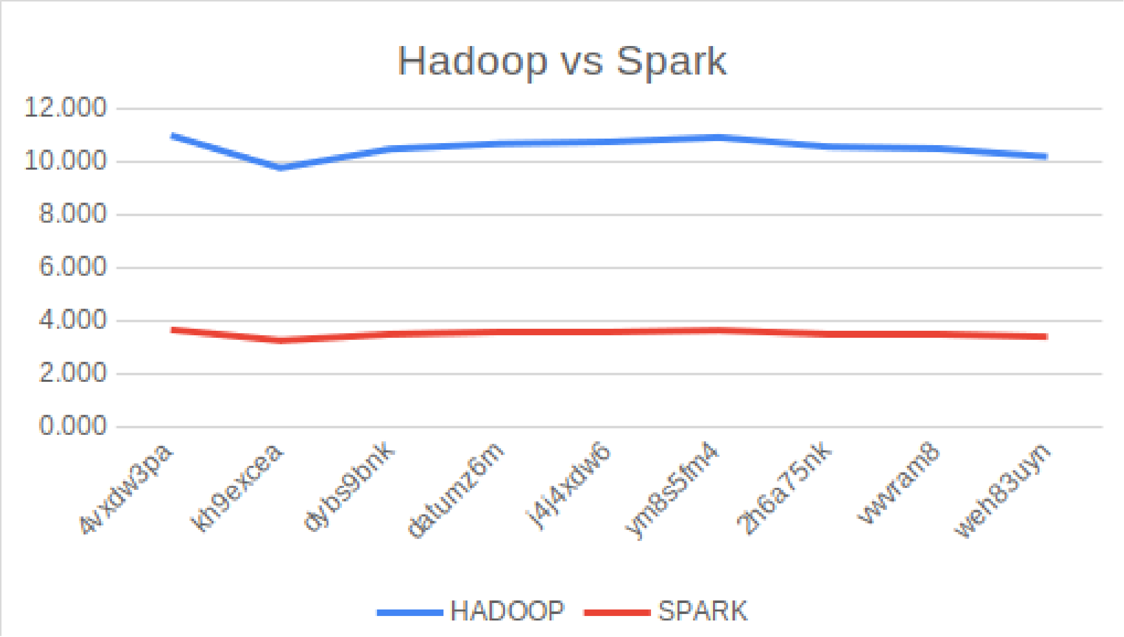


Figure 20: Hadoop vs Spark Comparison. User Time in seconds (2)

4 The social network problem

4.1 Describe how you have used MapReduce jobs to solve the social network problem.

- We have written two Python scripts: ‘mapper.py’ and ‘reducer.py’.
- mapper.py: Reads the source text file line by line. For each line in the source file which is in the format (UserID)(tab)(FriendsID), mapper outputs: (UserID)(tab)(FriendID)(‘friend’). For each FriendID in UserID’s friends list, it generates a combination in this format (FriendID)(tab)(OtherFriendID)(’not friend’).
- reducer.py: Reads the outputs of mapper.py line by line. Since we set UserID as key, all UserID’s data goes to the same reducer. Each reducer, counts the and relations of the users that it reads.

4.2 Describe your algorithm to tackle the social network problem.

- Our idea is to use UserID’s as keys and FriendID information as values. For example, if we have user 0 which is friends with users 2, 3, 4, and 5, the output of the mapper will be:
  - 0 2 friend
  - 0 3 friend
  - 0 4 friend
  - 0 5 friend
  - 2 3 not friend
  - 2 4 not friend
  - 2 5 not friend
  - 3 4 not friend
  - 3 5 not friend
  - 4 5 not friend

  Since friendships have been defined as both ways (if 0 is friends with 1, 1 is friends with 0), we sort the users by numerical order before generating outputs. This means that we will be able to have consistent keys for each user and send all that user’s information to the same reducer. The reducers read the outputs of mappers. Since we are using the UserID as key and we are sorting by numerical value, our reducers have the entire information for that UserID.
- Each reducer contains a dictionary keyed by UserIDs. Each UserID’s dictionary is a nested dictionary that contains the number of times each User has seen other users. If a relation of ‘friend’ has been seen between users the relation flag will be set to True and this user will not be counted for recommendation. If the ‘not friend’ is seen for two users, we will consider them as recommendation candidates. Finally, we filter the user’s dictionary by ’not friend’ flag to see which users should be recommended as friends.

- We tried multiple times with ideas inspired from the WordCount example. We first tried setting the key as UserID and FriendID. For the same example as above, the output would be:

  - (0 2) friend
  - (0 3) friend
  - (0 4) friend
  - (0 5) friend
  - (2 3) not friend
  - ....

  Like our final solution, these keys would be sorted numerically. However, even though this algorithm worked on a single machine, on a distributed system, it meant that not all users' information would go to the same reducer and thus reducers would have incomplete information.

- The other solution that we tried was the same as our current solution but without sorting the friend lists numerically when we were generating mapper's outputs. This solution counted duplicates for the same keys and was not able to find correct recomendations.

4.3 Presents your recommendations of connection for the users with following user IDs: 924, 8941, 8942, 9019, 9020, 9021, 9022, 9990, 9992, 9993

For each user, each recommendation contains two numbers. The first is the other UserID and the second is the number of mutual friends they have. Our results are presented below:

- User 924: (6995, 1) - (439, 1) - (2409, 1) - (11860, 1) - (15416, 1) - (43748, 1) - (45881, 1)
- User 8941: (8943, 2) - (8944, 2) - (8940, 1)
- User 8942: (8939, 3) - (8940, 1) - (8943, 1) - (8944, 1)
- User 9019: (9022, 2) - (317, 1) - (9023, 1)
- User 9020: (9021, 3) - (9016, 2) - (9017, 2) - (9022, 2) - (317, 1) - (9023, 1)
- User 9021: (9020, 3) - (9016, 2) - (9017, 2) - (9022, 2) - (317, 1) - (9023, 1)
- User 9022: (9019, 2) - (9020, 2) - (9021, 2) - (317, 1) - (9016, 1) - (9017, 1) - (9023, 1)
- User 9990: (13134, 1) - (13478, 1) - (13877, 1) - (34299, 1) - (34485, 1) - (34642, 1) - (37941, 1)
- User 9992: (9987, 4) - (9989, 4) - (35667, 3) - (9991, 2)
- User 9993: (9991, 5) - (13134, 1) - (13478, 1) - (13877, 1) - (34299, 1) - (34485, 1) - (34642, 1) - (37941, 1)

5 Summary of results and instructions to run your code

This section deals with steps to follow in order to reproduce our work (**we uploaded everything at our GitHub repository**[4]). First, we need to follow the instructions in the environment setup section 1 to install Hadoop and Spark adequately. Next, we should start the Apache Hadoop Cluster using the next commands:

```
$ start-dfs.sh
$ start-yarn.sh
$ jps # to verify all the running components
```

After that, Spark script can retrieve data from the input folder of HDFS and run the test.

For the first scenario which concerns comparing Hadoop to Linux, we should download our Github repo [4]. Then, we copy this script and the 9 datasets to the directory **/usr/local/hadoop/share /hadoop/mapreduce**. After that, we simply run the former script three times and get the average user time. We can see the results in this Excel file. At this stage, we can easily perform the comparison between Hadoop and Linux. For the second scenario which concerns comparing Hadoop to Spark, this file contains the comparison results, which shows that Spark has a much better performance. As for the social network scenario, before running the scripts, we needed to install Hadoop on our Azure VM [5]. After that, since we have used python instead of Java, we have used the instructions from this source [11]. In short, after installing Hadoop, first put the source file in Hadoop's HDFS directory. Then, we leverage Hadoop's streaming API to pass files between 'mapper.py' and 'reducer.py' by STDIN and STDOUT.

```
azureuser@ubuntu:/usr/local/hadoop
$ bin/hadoop jar contrib/streaming/hadoop-*streaming*.jar \
-mapper /home/azureuser/mapper.py \
-reducer /home/azureuser/reducer.py \
-input /user/azureuser/recom_example/cc_tp2.txt \
-output /user/azureuser/recome_output
```

References

[1] *BASIC HDFS FILE OPERATIONS COMMANDS*. URL: https://www.alluxio.io/learn/hdfs/basic-file-operations-commands/.

[2] *Command-line Tools can be 235x Faster than your Hadoop Cluster*. URL: https://adamdrake.com/command-line-tools-can-be-235x-faster-than-your-hadoop-cluster.html.

[3] *Command-line Tools vs Hadoop*. URL: https://www.reddit.com/r/programming/comments/8ljjzm/commandline_tools_can_be_235x_faster_than_your/.

[4] *Github Repo for Lab2*. URL: https://github.com/marsup13/CCF_Lab2_Fall2021.

[5] *Hadoop single node cluster setup*. URL: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html.

[6] *How to Install and Configure Hadoop on Ubuntu 20.04*. URL: https://tecadmin.net/install-hadoop-on-ubuntu-20-04/.

[7] *How to Install Hadoop on Ubuntu 18.04 or 20.04*. URL: https://phoenixnap.com/kb/install-hadoop-ubuntu.

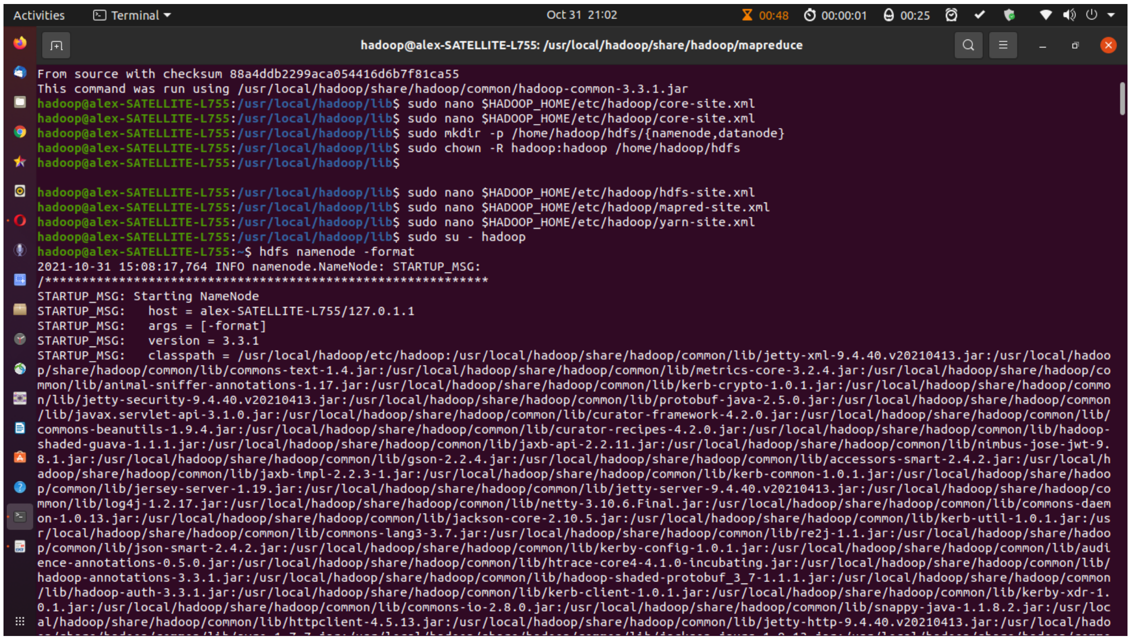
[8] *Install and Configure Apache Hadoop on Ubuntu 20.04*. URL: https://www.vultr.com/docs/install-and-configure-apache-hadoop-on-ubuntu-20-04.

[9] *RDD APIs*. URL: http://spark.apache.org/examples.html.

[10] *RDD programming guide*. URL: https://spark.apache.org/docs/latest/rdd-programming-guide.html#resilient-distributed-datasets-rdds.

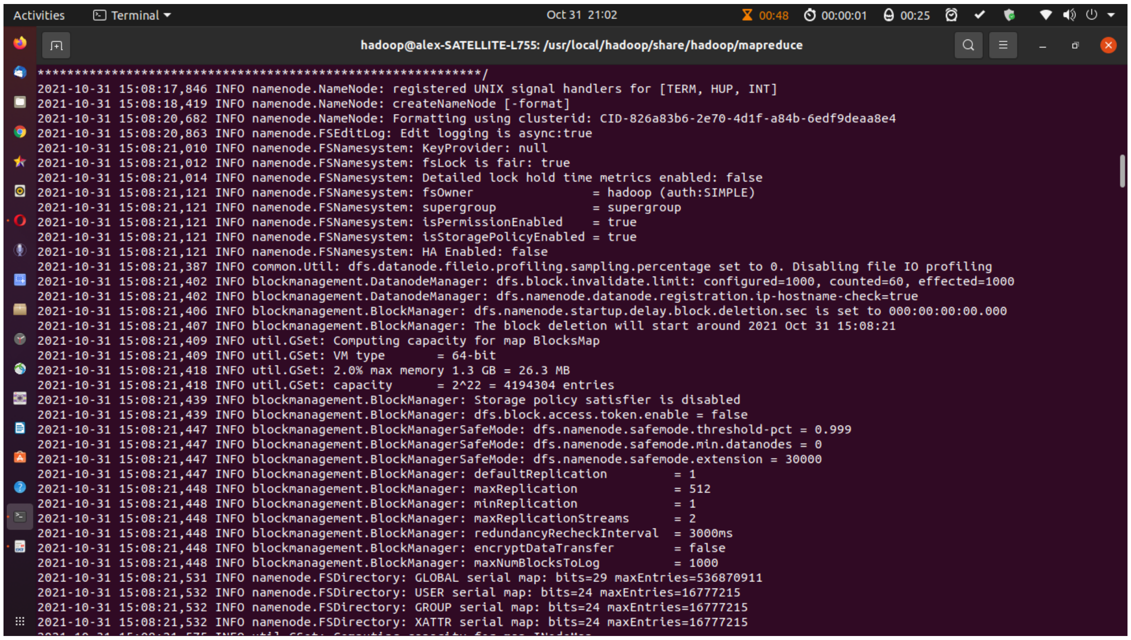
[11] *Writing An Hadoop MapReduce Program In Python*. URL: https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/.

A Appendix: Hadoop Configuration


Activities
Terminal
Oct 31 21:02
00:48
00:00:01
00:25
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

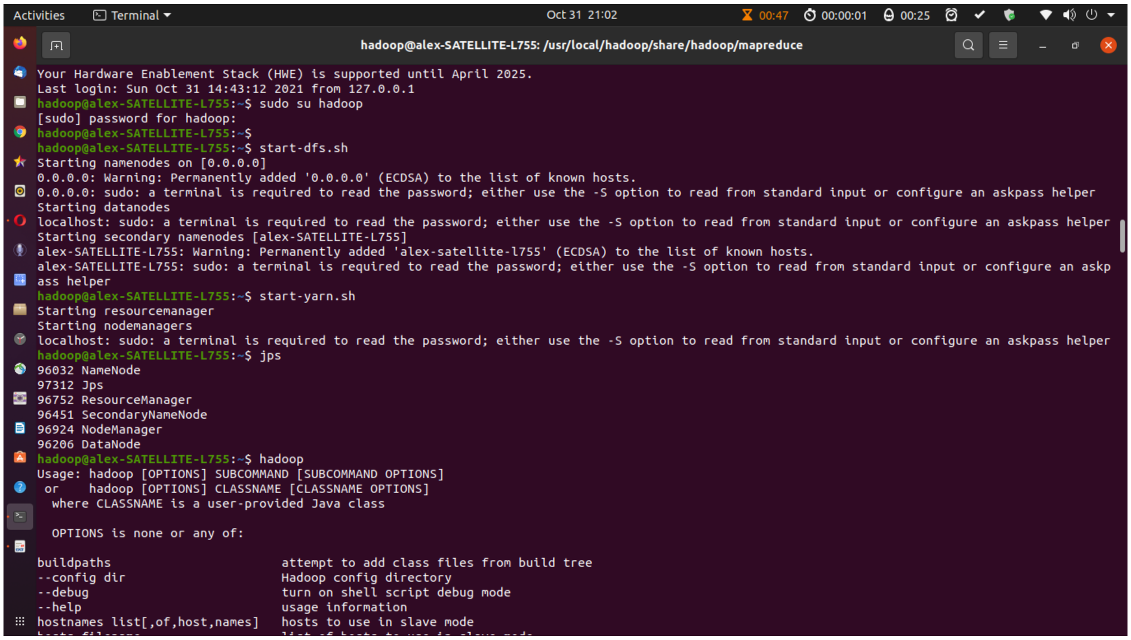
```
From source with checksum 88a4ddb2299aca054416d6b7f81ca55
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.1.jar
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo chown -R hadoop:hadoop /home/hadoop/hdfs
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$

hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/lib$ sudo su - hadoop
hadoop@alex-SATELLITE-L755:~$ hdfs namenode -format
2021-10-31 15:08:17,764 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = alex-SATELLITE-L755/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.1
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/jetty-xml-9.4.40.v20210413.jar:/usr/local/hadoo
p/share/hadoop/common/lib/commons-text-1.4.jar:/usr/local/hadoop/share/hadoop/common/lib/metrics-core-3.2.4.jar:/usr/local/hadoop/share/hadoop/co
mmon/lib/animal-sniffer-annotations-1.17.jar:/usr/local/hadoop/share/hadoop/common/lib/kerb-crypto-1.0.1.jar:/usr/local/hadoop/share/hadoop/commo
n/lib/jetty-security-9.4.40.v20210413.jar:/usr/local/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/common
/lib/javax.servlet-api-3.1.0.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-framework-4.2.0.jar:/usr/local/hadoop/share/hadoop/common/lib/
commons-beanutils-1.9.4.jar:/usr/local/hadoop/share/hadoop/common/lib/curator-recipes-4.2.0.jar:/usr/local/hadoop/share/hadoop/common/lib/hadoop-
shaded-guava-1.1.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jaxb-api-2.2.11.jar:/usr/local/hadoop/share/hadoop/common/lib/nimbus-jose-jwt-9.
8.1.jar:/usr/local/hadoop/share/hadoop/common/lib/gson-2.2.4.jar:/usr/local/hadoop/share/hadoop/common/lib/accessors-smart-2.4.2.jar:/usr/local/h
adoop/share/hadoop/common/lib/jaxb-impl-2.2.3-1.jar:/usr/local/hadoop/share/hadoop/common/lib/kerb-common-1.0.1.jar:/usr/local/hadoop/share/hadoo
p/common/lib/jersey-server-1.19.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-server-9.4.40.v20210413.jar:/usr/local/hadoop/share/hadoop/co
mmon/lib/log4j-1.2.17.jar:/usr/local/hadoop/share/hadoop/common/lib/netty-3.10.6.Final.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-daem
on-1.0.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-core-2.10.5.jar:/usr/local/hadoop/share/hadoop/common/lib/kerb-util-1.0.1.jar:/us
r/local/hadoop/share/hadoop/common/lib/commons-lang3-3.7.jar:/usr/local/hadoop/share/hadoop/common/lib/re2j-1.1.jar:/usr/local/hadoop/share/hadoo
p/common/lib/json-smart-2.4.2.jar:/usr/local/hadoop/share/hadoop/common/lib/kerby-config-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/audi
ence-annotations-0.5.0.jar:/usr/local/hadoop/share/hadoop/common/lib/htrace-core4-4.1.0-incubating.jar:/usr/local/hadoop/share/hadoop/common/lib/
hadoop-annotations-3.3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/hadoop-shaded-protobuf_3_7-1.1.1.jar:/usr/local/hadoop/share/hadoop/common
/lib/hadoop-auth-3.3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/kerb-client-1.0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/kerby-xdr-1.
0.1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-io-2.8.0.jar:/usr/local/hadoop/share/hadoop/common/lib/snappy-java-1.1.8.2.jar:/usr/loc
al/hadoop/share/hadoop/common/lib/httpclient-4.5.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jetty-http-9.4.40.v20210413.jar:/usr/local/hado
```

Figure 21: Testing Hadoop (1)


Activities
Terminal
Oct 31 21:02
00:48
00:00:01
00:25
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

```
************************************************************/
2021-10-31 15:08:17,846 INFO namenode.NameNode: registered UNIX signal handlers for [TERM, HUP, INT]
2021-10-31 15:08:18,419 INFO namenode.NameNode: createNameNode [-format]
2021-10-31 15:08:20,682 INFO namenode.NameNode: Formatting using clusterid: CID-826a83b6-2e70-4d1f-a84b-6edf9deaa8e4
2021-10-31 15:08:20,863 INFO namenode.FSEditLog: Edit logging is async:true
2021-10-31 15:08:21,010 INFO namenode.FSNamesystem: KeyProvider: null
2021-10-31 15:08:21,012 INFO namenode.FSNamesystem: fsLock is fair: true
2021-10-31 15:08:21,014 INFO namenode.FSNamesystem: Detailed lock hold time metrics enabled: false
2021-10-31 15:08:21,121 INFO namenode.FSNamesystem: fsOwner                = hadoop (auth:SIMPLE)
2021-10-31 15:08:21,121 INFO namenode.FSNamesystem: supergroup             = supergroup
2021-10-31 15:08:21,121 INFO namenode.FSNamesystem: isPermissionEnabled    = true
2021-10-31 15:08:21,121 INFO namenode.FSNamesystem: isStoragePolicyEnabled = true
2021-10-31 15:08:21,121 INFO namenode.FSNamesystem: HA Enabled: false
2021-10-31 15:08:21,387 INFO common.Util: dfs.datanode.fileio.profiling.sampling.percentage set to 0. Disabling file IO profiling
2021-10-31 15:08:21,402 INFO blockmanagement.DatanodeManager: dfs.block.invalidate.limit: configured=1000, counted=60, effected=1000
2021-10-31 15:08:21,402 INFO blockmanagement.DatanodeManager: dfs.namenode.datanode.registration.ip-hostname-check=true
2021-10-31 15:08:21,406 INFO blockmanagement.BlockManager: dfs.namenode.startup.delay.block.deletion.sec is set to 000:00:00:00.000
2021-10-31 15:08:21,407 INFO blockmanagement.BlockManager: The block deletion will start around 2021 Oct 31 15:08:21
2021-10-31 15:08:21,409 INFO util.GSet: Computing capacity for map BlocksMap
2021-10-31 15:08:21,409 INFO util.GSet: VM type       = 64-bit
2021-10-31 15:08:21,418 INFO util.GSet: 2.0% max memory 1.3 GB = 26.3 MB
2021-10-31 15:08:21,418 INFO util.GSet: capacity      = 2^22 = 4194304 entries
2021-10-31 15:08:21,439 INFO blockmanagement.BlockManager: Storage policy satisfier is disabled
2021-10-31 15:08:21,439 INFO blockmanagement.BlockManager: dfs.block.access.token.enable = false
2021-10-31 15:08:21,447 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.threshold-pct = 0.999
2021-10-31 15:08:21,447 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.min.datanodes = 0
2021-10-31 15:08:21,447 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.extension = 30000
2021-10-31 15:08:21,447 INFO blockmanagement.BlockManager: defaultReplication         = 1
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: maxReplication             = 512
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: minReplication             = 1
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: maxReplicationStreams      = 2
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: redundancyRecheckInterval  = 3000ms
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: encryptDataTransfer        = false
2021-10-31 15:08:21,448 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
2021-10-31 15:08:21,531 INFO namenode.FSDirectory: GLOBAL serial map: bits=29 maxEntries=536870911
2021-10-31 15:08:21,532 INFO namenode.FSDirectory: USER serial map: bits=24 maxEntries=16777215
2021-10-31 15:08:21,532 INFO namenode.FSDirectory: GROUP serial map: bits=24 maxEntries=16777215
2021-10-31 15:08:21,532 INFO namenode.FSDirectory: XATTR serial map: bits=24 maxEntries=16777215
```

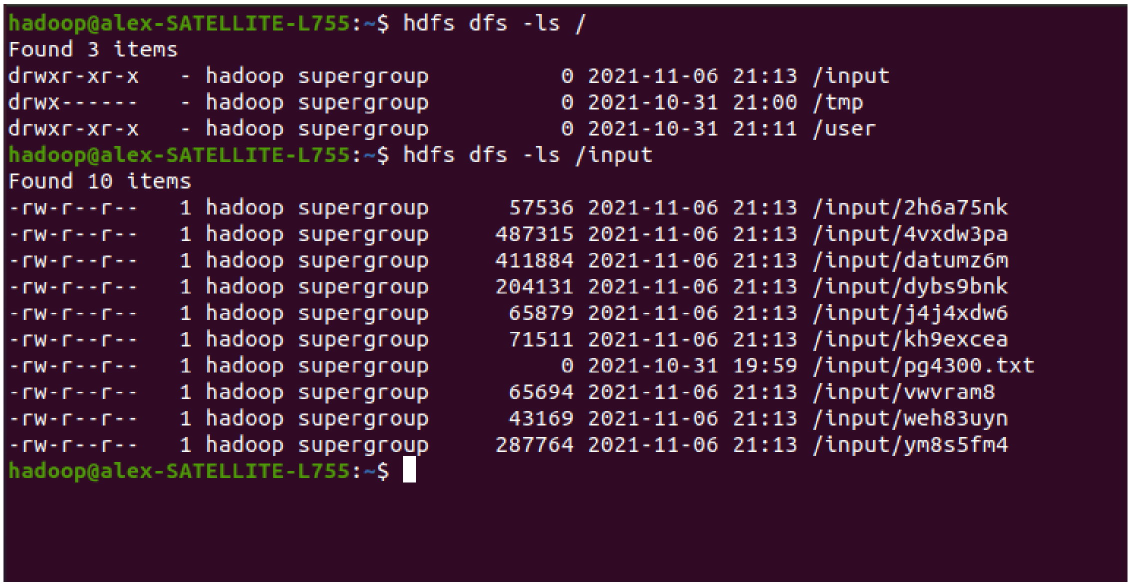
Figure 22: Testing Hadoop (2)


Activities
Terminal
Oct 31 21:02
00:47
00:00:01
00:25
hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce

```
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct 31 14:43:12 2021 from 127.0.0.1
hadoop@alex-SATELLITE-L755:~$ sudo su hadoop
[sudo] password for hadoop:
hadoop@alex-SATELLITE-L755:~$
hadoop@alex-SATELLITE-L755:~$ start-dfs.sh
Starting namenodes on [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
Starting datanodes
localhost: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
Starting secondary namenodes [alex-SATELLITE-L755]
alex-SATELLITE-L755: Warning: Permanently added 'alex-satellite-l755' (ECDSA) to the list of known hosts.
alex-SATELLITE-L755: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askp
ass helper
hadoop@alex-SATELLITE-L755:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
localhost: sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
hadoop@alex-SATELLITE-L755:~$ jps
96032 NameNode
97312 Jps
96752 ResourceManager
96451 SecondaryNameNode
96924 NodeManager
96206 DataNode
hadoop@alex-SATELLITE-L755:~$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
 or    hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
  where CLASSNAME is a user-provided Java class

  OPTIONS is none or any of:

buildpaths                       attempt to add class files from build tree
--config dir                     Hadoop config directory
--debug                          turn on shell script debug mode
--help                           usage information
hostnames list[,of,host,names]   hosts to use in slave mode
```

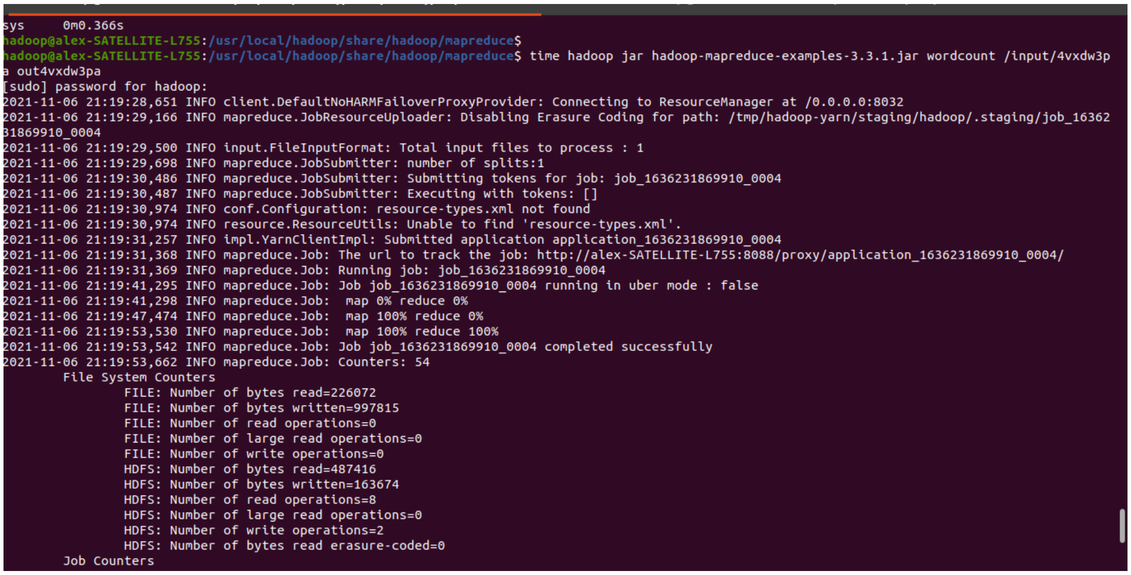
Figure 23: Testing Hadoop (3)

B Appendix: Hadoop vs Linux Comparison



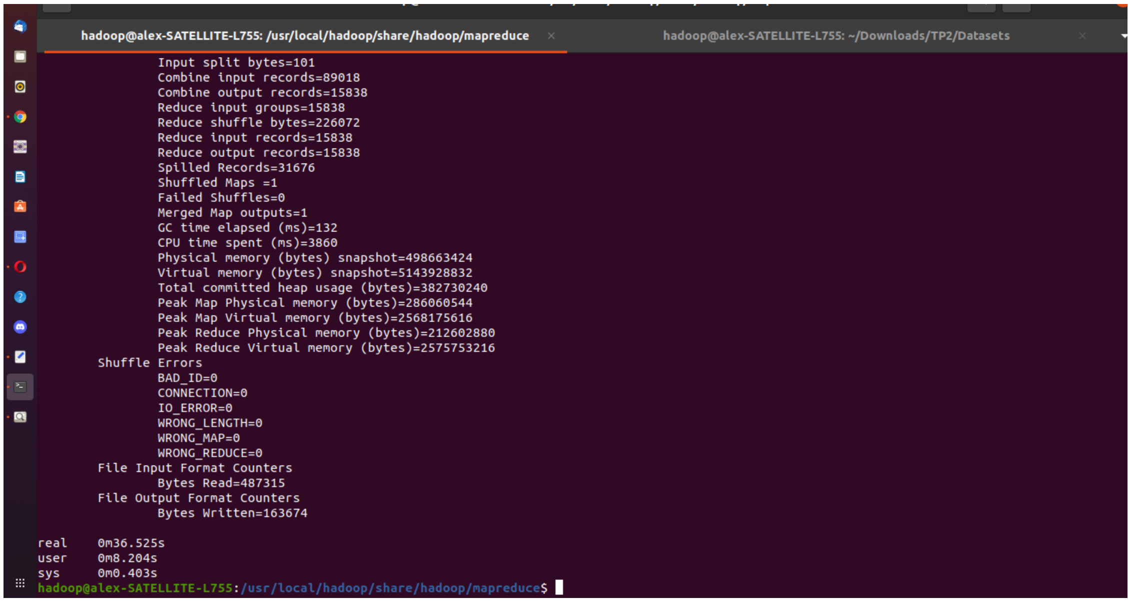
```
hadoop@alex-SATELLITE-L755:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2021-11-06 21:13 /input
drwx------   - hadoop supergroup          0 2021-10-31 21:00 /tmp
drwxr-xr-x   - hadoop supergroup          0 2021-10-31 21:11 /user
hadoop@alex-SATELLITE-L755:~$ hdfs dfs -ls /input
Found 10 items
-rw-r--r--   1 hadoop supergroup      57536 2021-11-06 21:13 /input/2h6a75nk
-rw-r--r--   1 hadoop supergroup     487315 2021-11-06 21:13 /input/4vxdw3pa
-rw-r--r--   1 hadoop supergroup     411884 2021-11-06 21:13 /input/datumz6m
-rw-r--r--   1 hadoop supergroup     204131 2021-11-06 21:13 /input/dybs9bnk
-rw-r--r--   1 hadoop supergroup      65879 2021-11-06 21:13 /input/j4j4xdw6
-rw-r--r--   1 hadoop supergroup      71511 2021-11-06 21:13 /input/kh9excea
-rw-r--r--   1 hadoop supergroup          0 2021-10-31 19:59 /input/pg4300.txt
-rw-r--r--   1 hadoop supergroup      65694 2021-11-06 21:13 /input/vwvram8
-rw-r--r--   1 hadoop supergroup      43169 2021-11-06 21:13 /input/weh83uyn
-rw-r--r--   1 hadoop supergroup     287764 2021-11-06 21:13 /input/ym8s5fm4
hadoop@alex-SATELLITE-L755:~$
```

Figure 24: Looking through all datasets



```
sys     0m0.366s
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$ time hadoop jar hadoop-mapreduce-examples-3.3.1.jar wordcount /input/4vxdw3p
a out4vxdw3pa
[sudo] password for hadoop:
2021-11-06 21:19:28,651 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-06 21:19:29,166 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_16362
31869910_0004
2021-11-06 21:19:29,500 INFO input.FileInputFormat: Total input files to process : 1
2021-11-06 21:19:29,698 INFO mapreduce.JobSubmitter: number of splits:1
2021-11-06 21:19:30,486 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1636231869910_0004
2021-11-06 21:19:30,487 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-06 21:19:30,974 INFO conf.Configuration: resource-types.xml not found
2021-11-06 21:19:30,974 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-11-06 21:19:31,257 INFO impl.YarnClientImpl: Submitted application application_1636231869910_0004
2021-11-06 21:19:31,368 INFO mapreduce.Job: The url to track the job: http://alex-SATELLITE-L755:8088/proxy/application_1636231869910_0004/
2021-11-06 21:19:31,369 INFO mapreduce.Job: Running job: job_1636231869910_0004
2021-11-06 21:19:41,295 INFO mapreduce.Job: Job job_1636231869910_0004 running in uber mode : false
2021-11-06 21:19:41,298 INFO mapreduce.Job:  map 0% reduce 0%
2021-11-06 21:19:47,474 INFO mapreduce.Job:  map 100% reduce 0%
2021-11-06 21:19:53,530 INFO mapreduce.Job:  map 100% reduce 100%
2021-11-06 21:19:53,542 INFO mapreduce.Job: Job job_1636231869910_0004 completed successfully
2021-11-06 21:19:53,662 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=226072
                FILE: Number of bytes written=997815
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=487416
                HDFS: Number of bytes written=163674
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
```

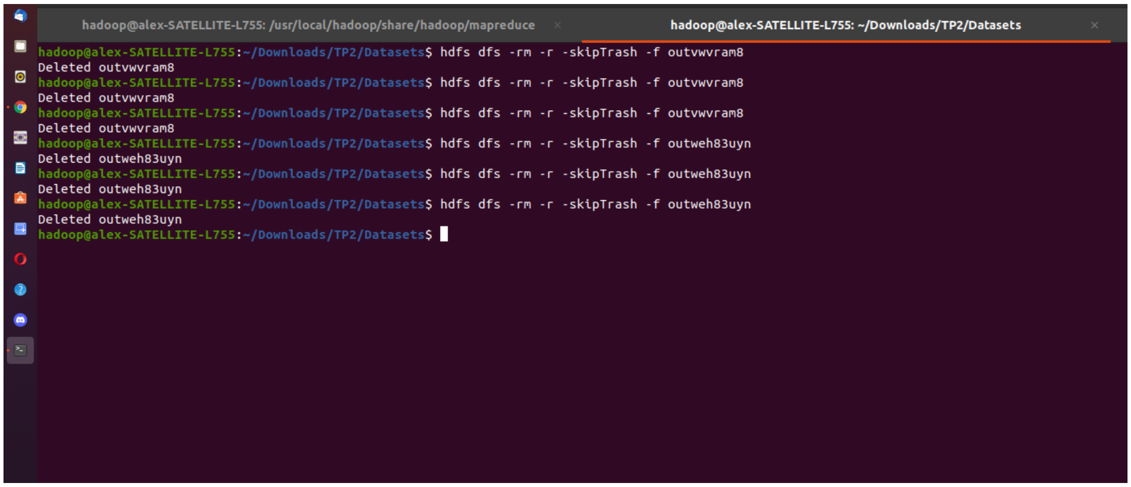
Figure 25: Measuring the execution time of dataset 1



hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
hadoop@alex-SATELLITE-L755: ~/Downloads/TP2/Datasets

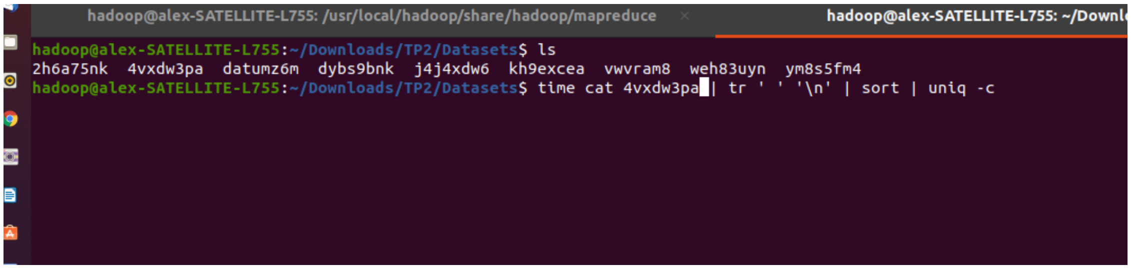
```
                Input split bytes=101
                Combine input records=89018
                Combine output records=15838
                Reduce input groups=15838
                Reduce shuffle bytes=226072
                Reduce input records=15838
                Reduce output records=15838
                Spilled Records=31676
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=132
                CPU time spent (ms)=3860
                Physical memory (bytes) snapshot=498663424
                Virtual memory (bytes) snapshot=5143928832
                Total committed heap usage (bytes)=382730240
                Peak Map Physical memory (bytes)=286060544
                Peak Map Virtual memory (bytes)=2568175616
                Peak Reduce Physical memory (bytes)=212602880
                Peak Reduce Virtual memory (bytes)=2575753216
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=487315
        File Output Format Counters
                Bytes Written=163674

real    0m36.525s
user    0m8.204s
sys     0m0.403s
hadoop@alex-SATELLITE-L755:/usr/local/hadoop/share/hadoop/mapreduce$
```

Figure 26: Measuring the execution time of dataset 1 (2)


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
hadoop@alex-SATELLITE-L755: ~/Downloads/TP2/Datasets

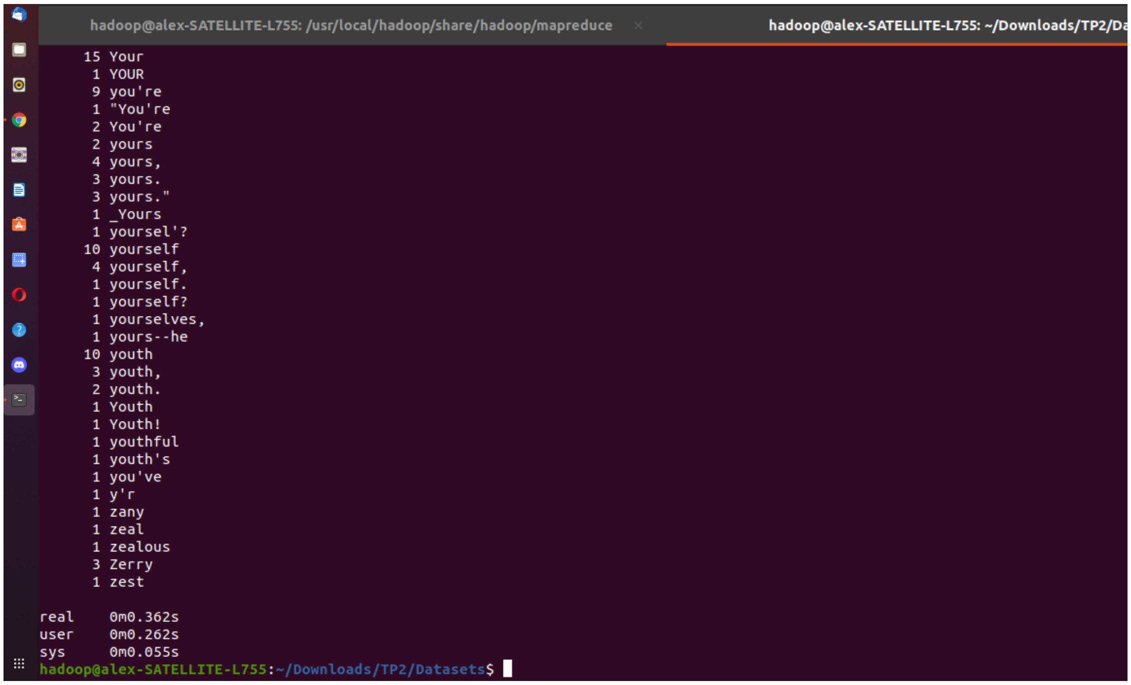
```
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outvwvram8
Deleted outvwvram8
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outvwvram8
Deleted outvwvram8
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outvwvram8
Deleted outvwvram8
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outweh83uyn
Deleted outweh83uyn
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outweh83uyn
Deleted outweh83uyn
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ hdfs dfs -rm -r -skipTrash -f outweh83uyn
Deleted outweh83uyn
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$
```

Figure 27: Deleting output after each experiment


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
hadoop@alex-SATELLITE-L755: ~/Downl

```
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ ls
2h6a75nk  4vxdw3pa  datumz6m  dybs9bnk  j4j4xdw6  kh9excea  vwvram8  weh83uyn  ym8s5fm4
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ time cat 4vxdw3pa| tr ' ' '\n' | sort | uniq -c
```

Figure 28: Using Linux command line to perform word frequencies of the dataset


hadoop@alex-SATELLITE-L755: /usr/local/hadoop/share/hadoop/mapreduce
hadoop@alex-SATELLITE-L755: ~/Downloads/TP2/Da

```
     15 Your
      1 YOUR
      9 you're
      1 "You're
      2 You're
      2 yours
      4 yours,
      3 yours.
      3 yours."
      1 _Yours
      1 yoursel'?
     10 yourself
      4 yourself,
      1 yourself.
      1 yourself?
      1 yourselves,
      1 yours--he
     10 youth
      3 youth,
      2 youth.
      1 Youth
      1 Youth!
      1 youthful
      1 youth's
      1 you've
      1 y'r
      1 zany
      1 zeal
      1 zealous
      3 Zerry
      1 zest

real    0m0.362s
user    0m0.262s
sys     0m0.055s
hadoop@alex-SATELLITE-L755:~/Downloads/TP2/Datasets$ 
```

Figure 29: Using Linux command line to perform word frequencies of the dataset (2)