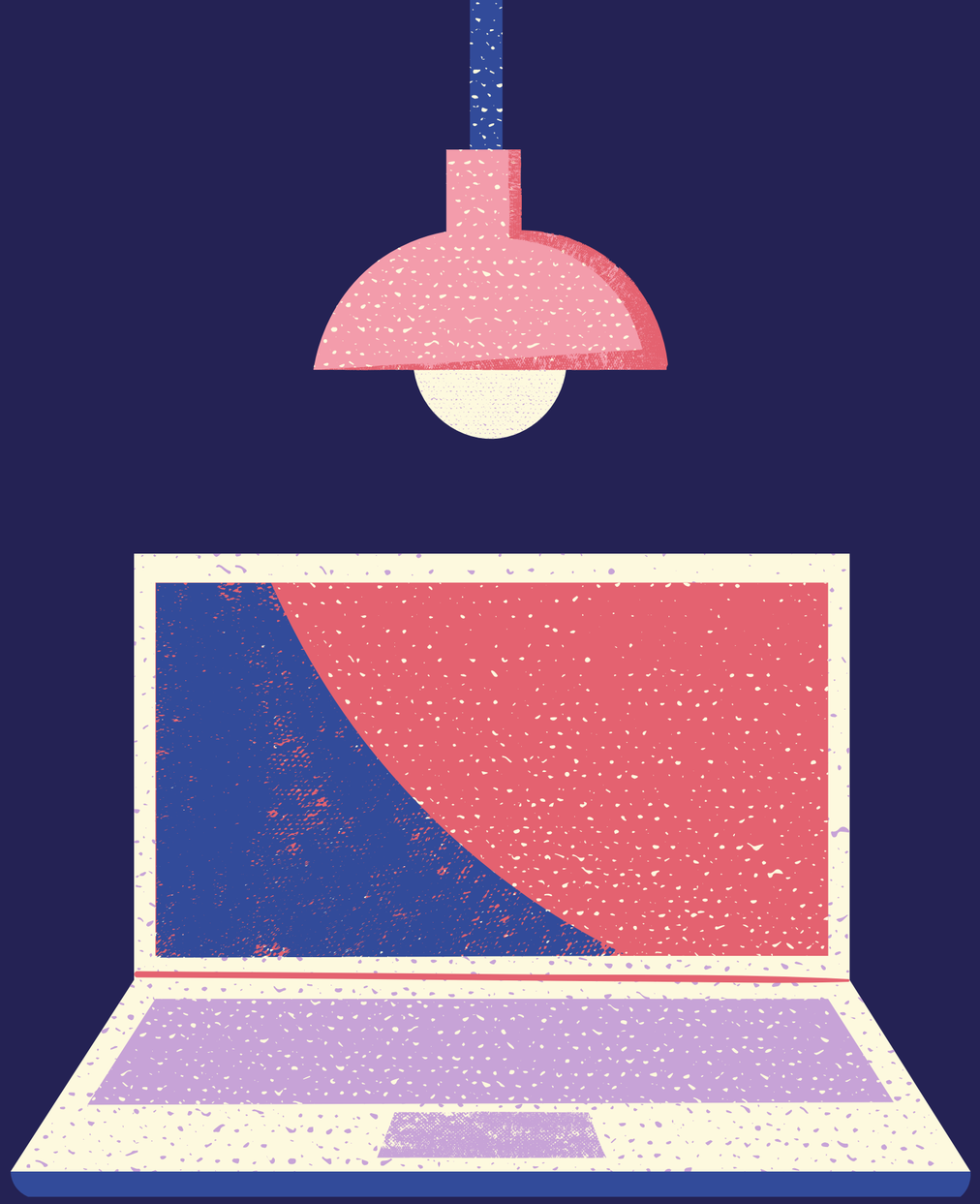
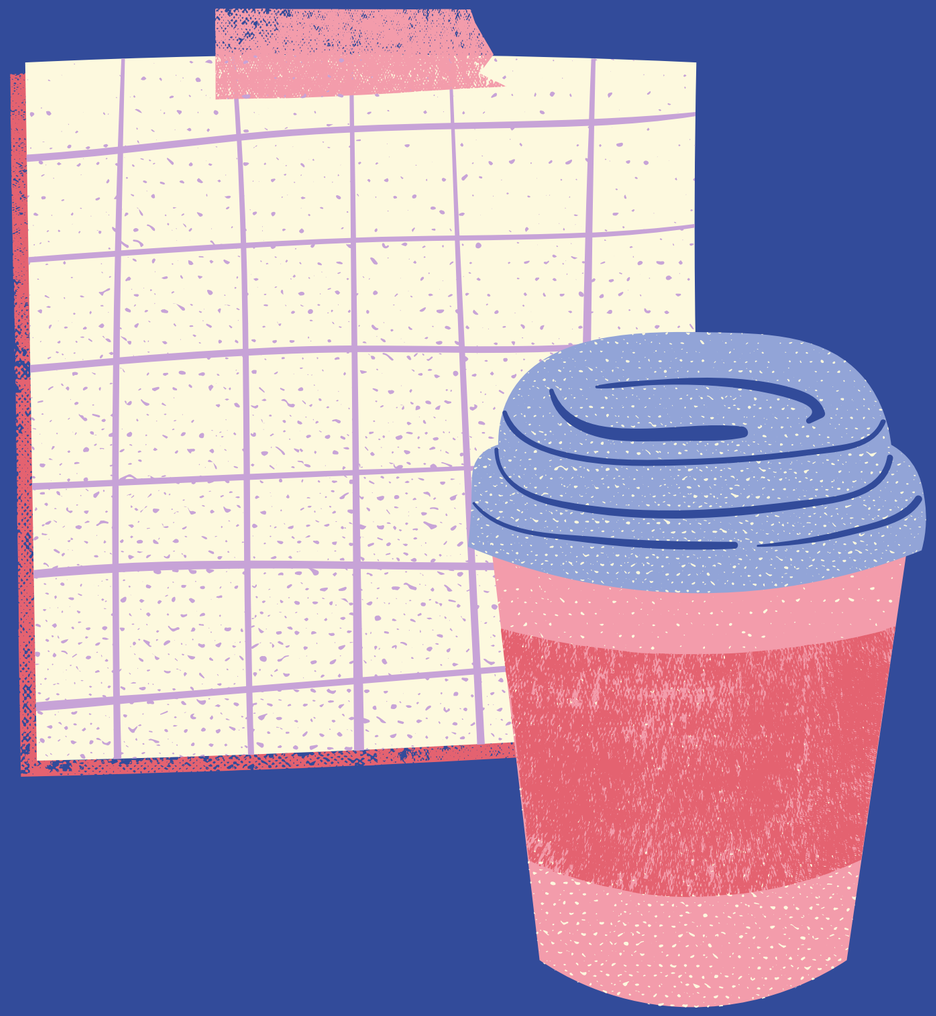


WORD COUNT PROGRAM

Hadoop vs Python

Group 3

- Cecilia Inez Reva Manurung (2106636994)
- Gemilang Bagas Ramadhani (2006535205)
- Laode Alif Ma'sum Sidrajat Raja Ika (2106731213)
- Zalfy Putra Rezky (2106731453)



Link Github

https://github.com/zalfyputra/hadoop-vs-python

Installing Hadoop on Ubuntu 20.04

https://medium.com/@festusmorumbasi/installing-hadoop-on-ubuntu-20-04-4610b6e0391e

Install Java 1.8

```
hadoop@vmbox:~$ sudo apt install openjdk-8-jdk -y
Reading package lists... Done
Building dependency tree
```

```
hadoop@vmbox:~$ java -version
openjdk version "1.8.0_362"
OpenJDK Runtime Environment (build 1.8.0_362-8u372-ga~us1-0ubuntu1~20.04-b09)
OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)
hadoop@vmbox:~$
```

Install Hadioop 3.3.2

```
hadoop@vmbox:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.tar.gz
--2023-06-21 13:40:45--  https://downloads.apache.org/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.tar
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.214.104, 2a01:4f9:
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 638660563 (609M) [application/x-gzip]
Saving to: 'hadoop-3.3.2.tar.gz'
```

Configure hadoop Env and XMLs

```
hadoop@vmbox:~$ sudo nano /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh
hadoop@vmbox:~$ sudo nano /home/hadoop/hadoop/etc/hadoop/core-site.xml
hadoop@vmbox:~$ sudo nano /home/hadoop/hadoop/etc/hadoop/hdfs-site.xml
hadoop@vmbox:~$ sudo nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
hadoop@vmbox:~$ sudo nano /home/hadoop/hadoop/etc/hadoop/yarn-site.xml
hadoop@vmbox:~$
```

Format HDFS NameNode

```
hadoop@vmbox:~/hadoop/etc/hadoop$ hdfs namenode -format
WARNING: /home/hadoop/hadoop/logs does not exist. Creating.
2023-06-21 15:07:31,421 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = vmbox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.2
```

Install and configure OpenSSH

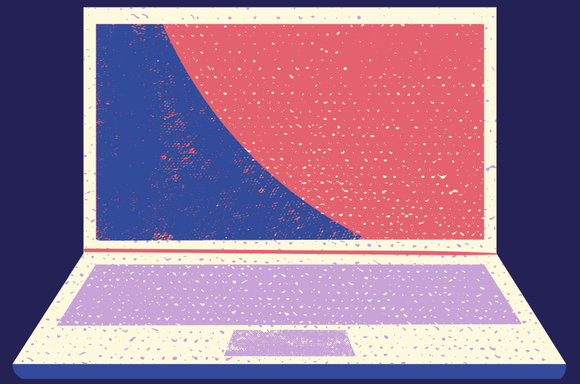
```
hadoop@vmbox:~/hadoop/etc/hadoop$ sudo apt install openssh-server openssh-client -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
hadoop@vmbox:~/hadoop/etc/hadoop$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:JW4j5U3zdHTCr8hp4vAySElV3OOS5Y9aI4fsdjwOEOU hadoop@vmbox
The key's randomart image is:
+---[RSA 3072]----+
|         oo. .o .|
|        .o. +..o |
|       .+ E=.... |
|      .+ *o+o.  .|
|     ...S..+.= . |
|      oo.o= X o  |
|     . . =.O .   |
|      . o *.+    |
|         + o..   |
+----[SHA256]-----+
hadoop@vmbox:~/hadoop/etc/hadoop$
```

```
hadoop@vmbox:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:QBcnDp2QJGAQydtFwoHHFS62ohvogzPncMXegvhpdgo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

2 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.2 LTS' available.
```

```
hadoop@vmbox:~$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hadoop@vmbox:~$ sudo chmod 640 ~/.ssh/authorized_keys
```

SPESIFIKASI KOMPUTER



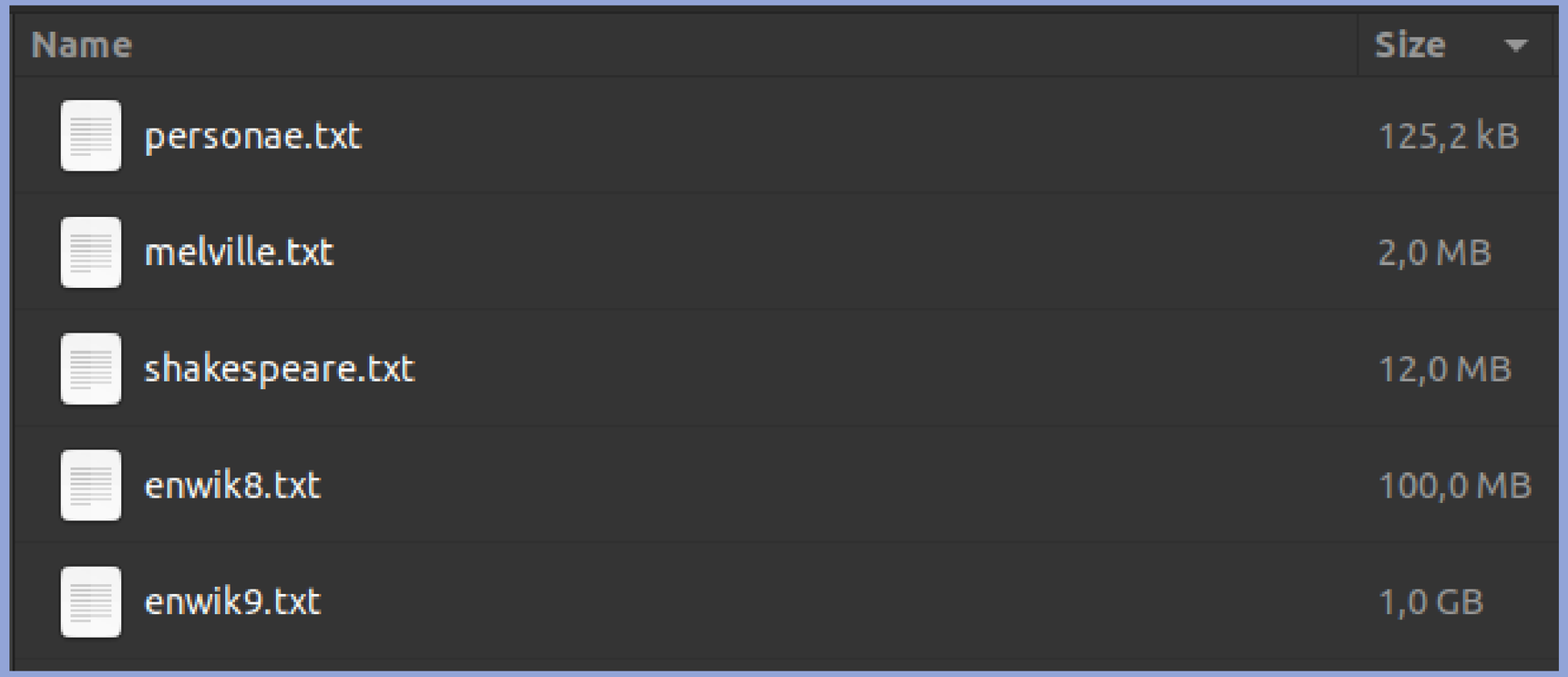
Host

CPU: AMD Ryzen 7 5800 3.2 GHz
GPU: NVIDIA GeForce RTX 3060
RAM: 32 GB
OS: Windows 11 Home 64-bit

VirtualBox

CPU: 8-core
RAM: 28 GB
OS: Ubuntu 20.04

File Test



References

- https://corpus.canterbury.ac.nz/descriptions/
- https://www.gutenberg.org/browse/scores/top
- https://mattmahoney.net/dc/textdata.html

Start Hadoop Server

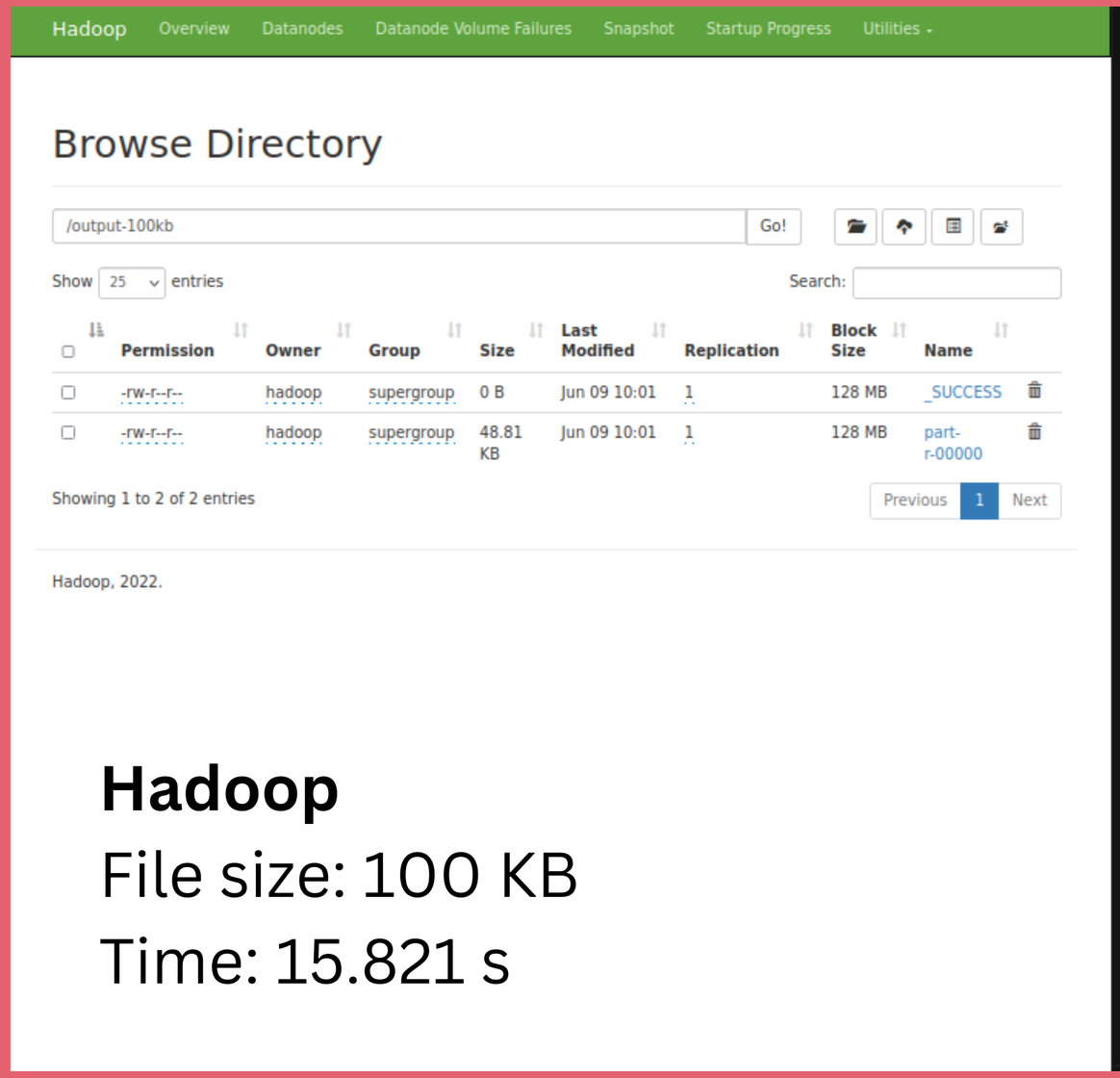
```
hadoop@vmbox:~$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [vmbox]
vmbox: Warning: Permanently added 'vmbox' (ECDSA) to the list of known hosts.
hadoop@vmbox:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@vmbox:~$ jps
7345 Jps
6392 DataNode
6828 ResourceManager
6990 NodeManager
6606 SecondaryNameNode
hadoop@vmbox:~$
```

Commands

1. Create a new folder in hdfs
2. Upload text file to the created folder
3. Run Hadoop MapReduce JAR file to count words of the text file
4. Store the output to a text file in local

example of 100 KB text file

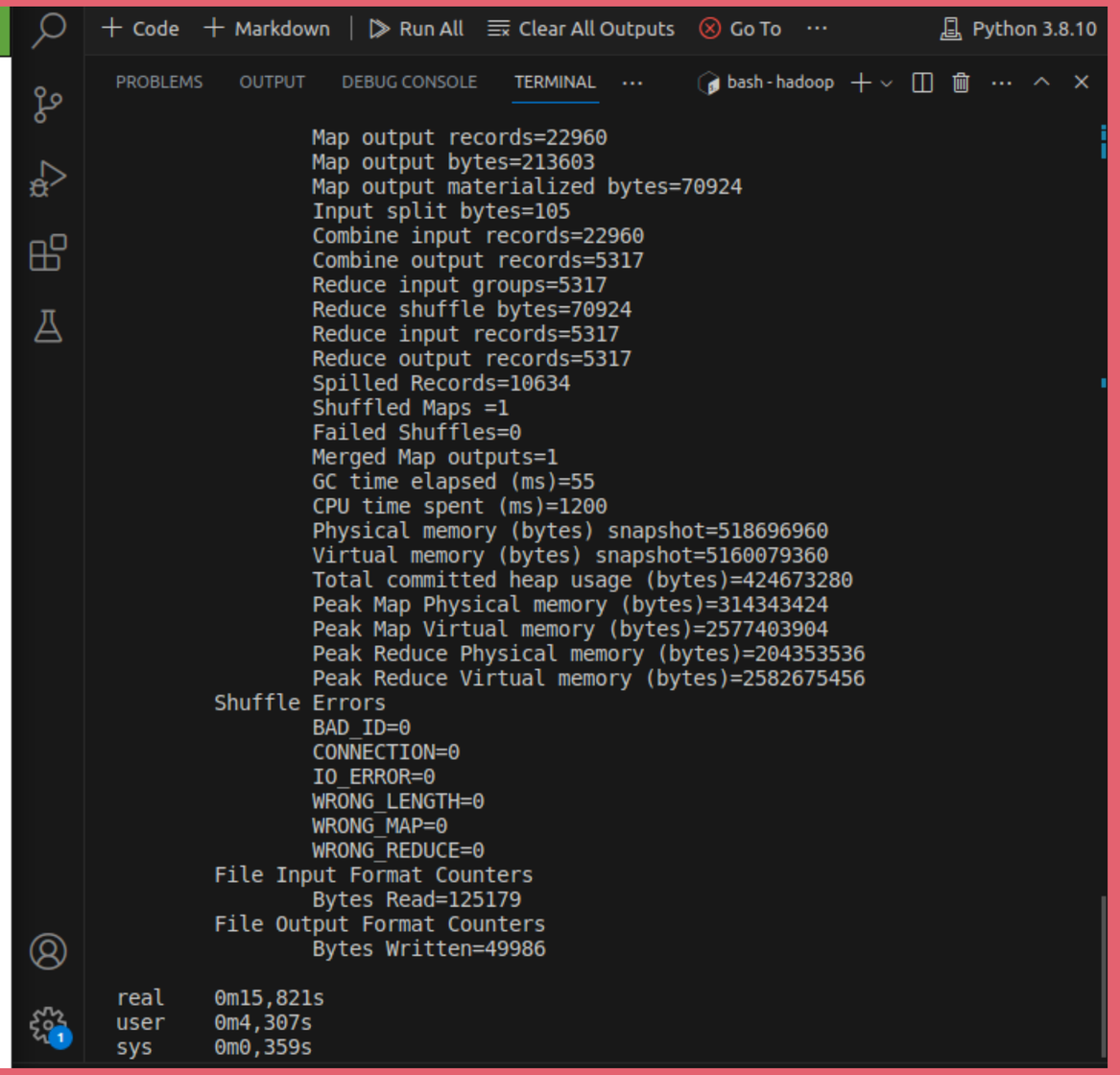
```
hadoop@vmbox:~$ hdfs dfs -mkdir /100kb
hadoop@vmbox:~$ hdfs dfs -put /home/hadoop/Downloads/wordcount/personae.txt /100kb
hadoop@vmbox:~$ time hadoop jar /home/hadoop/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.2.jar wordcount /100kb /
output-100kb
hadoop@vmbox:~$ hdfs dfs -get /output-100kb/part-r-00000 /home/hadoop/Downloads/wordcount/output-100kb.txt
```

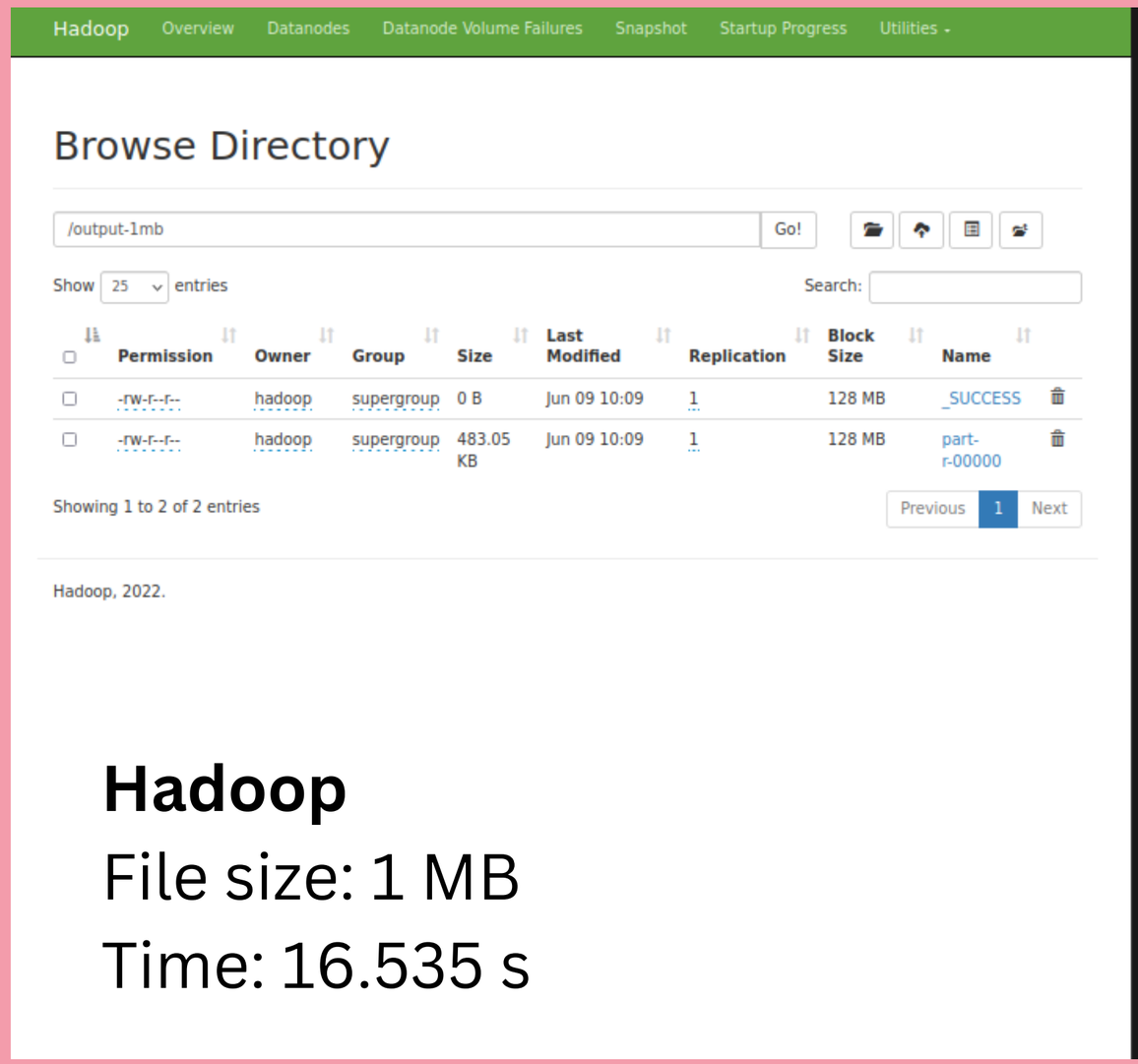


Hadoop

File size: 100 KB

Time: 15.821 s

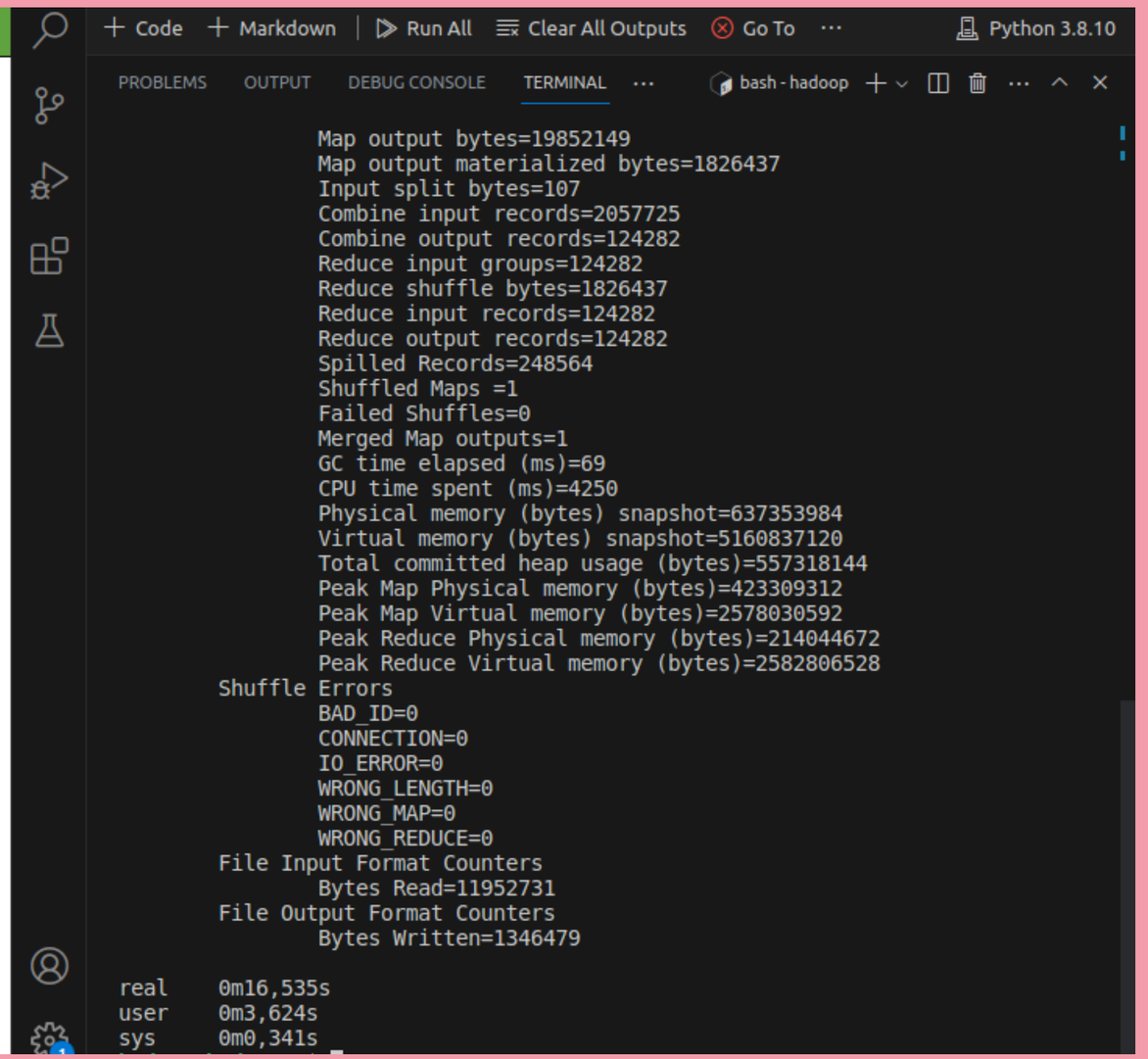


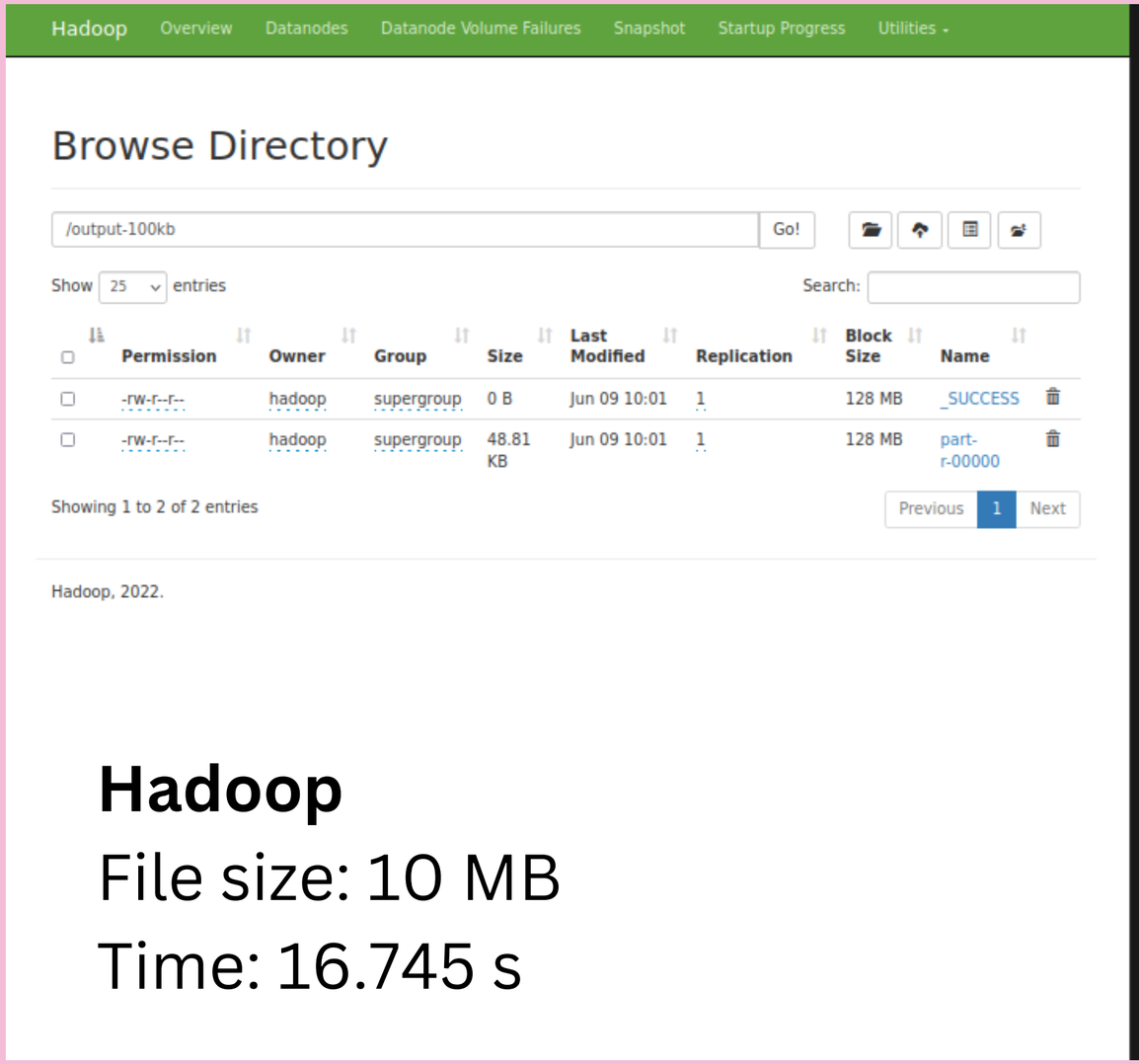


Hadoop

File size: 1 MB

Time: 16.535 s

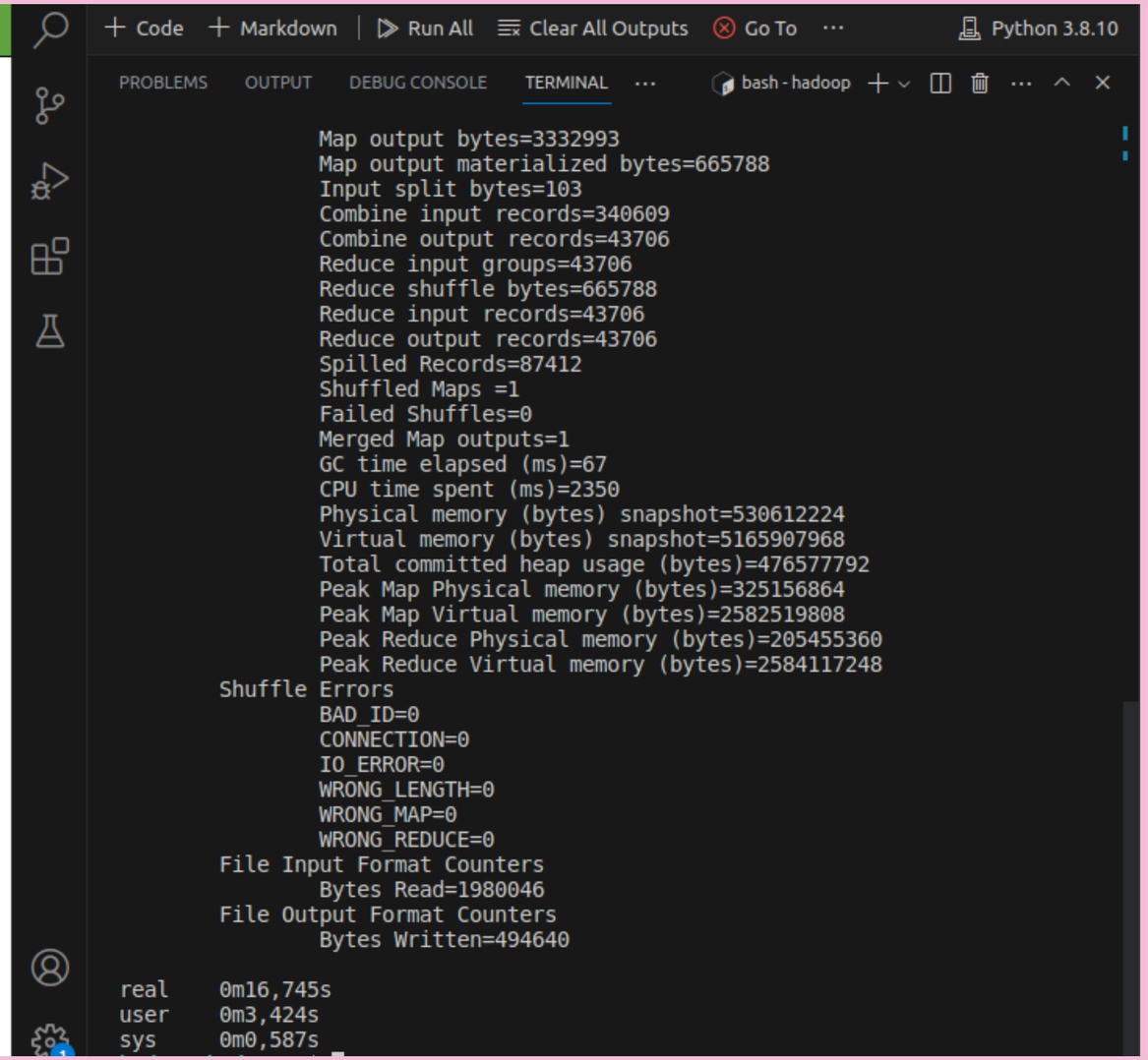


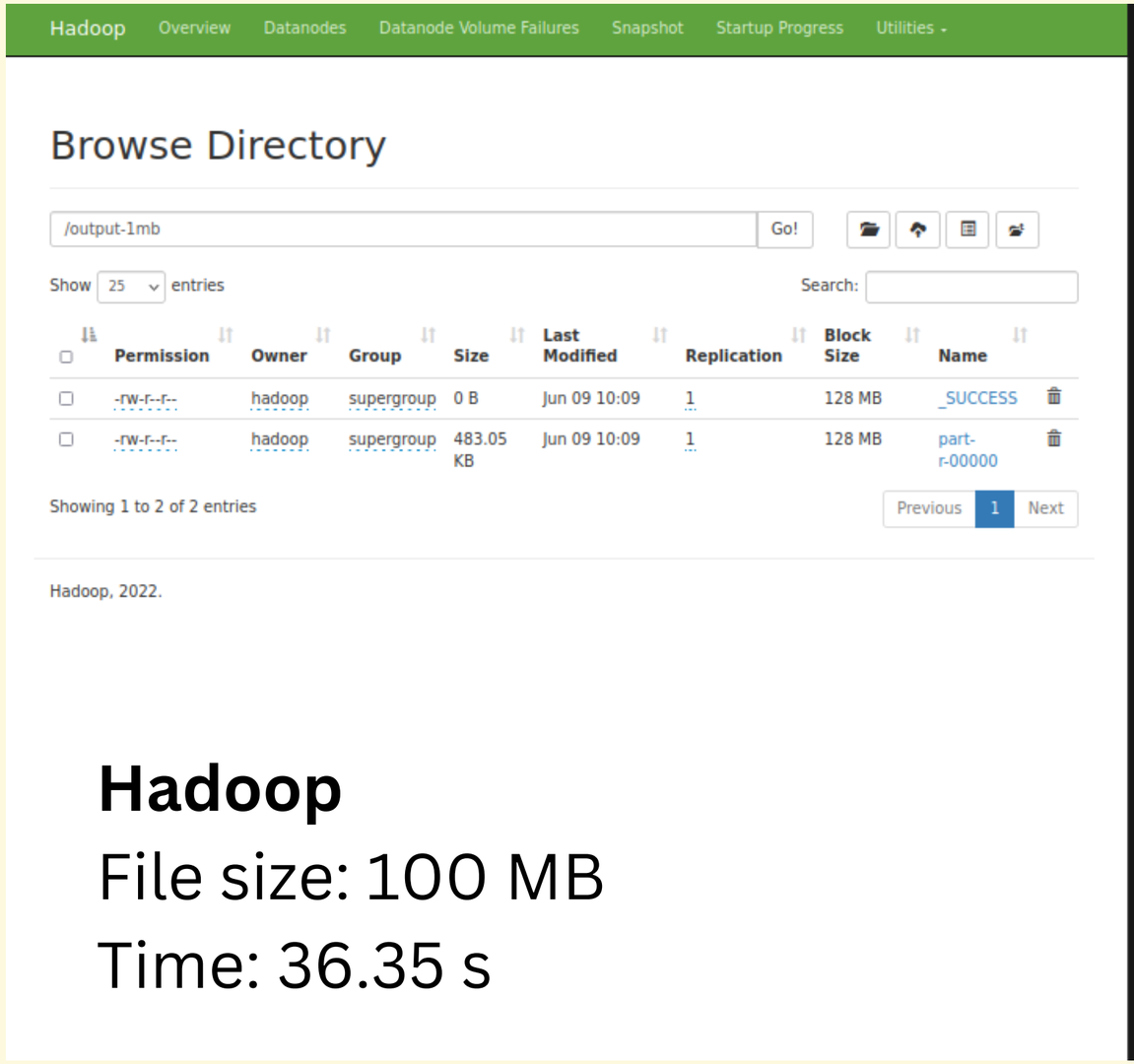


Hadoop

File size: 10 MB

Time: 16.745 s

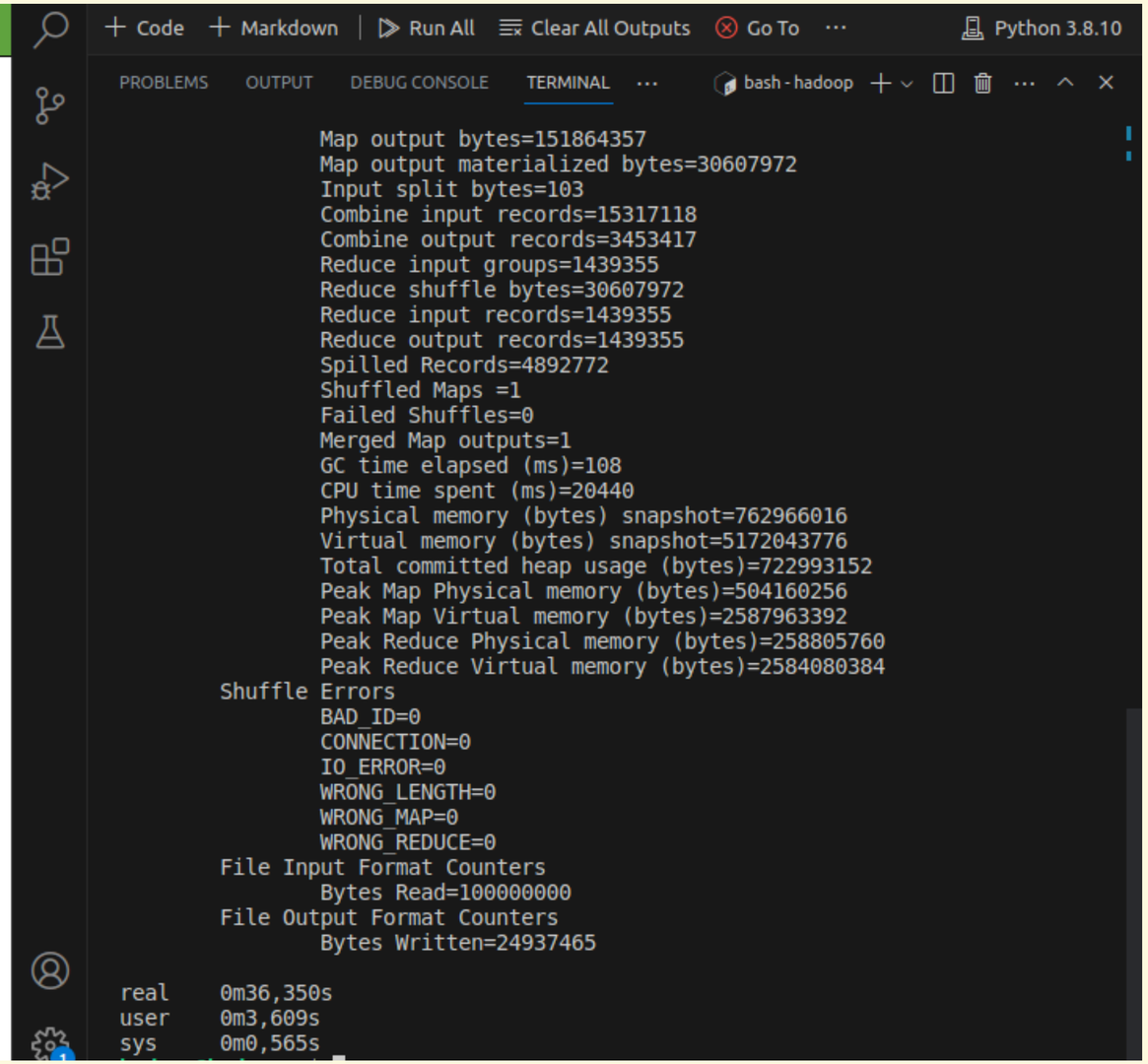


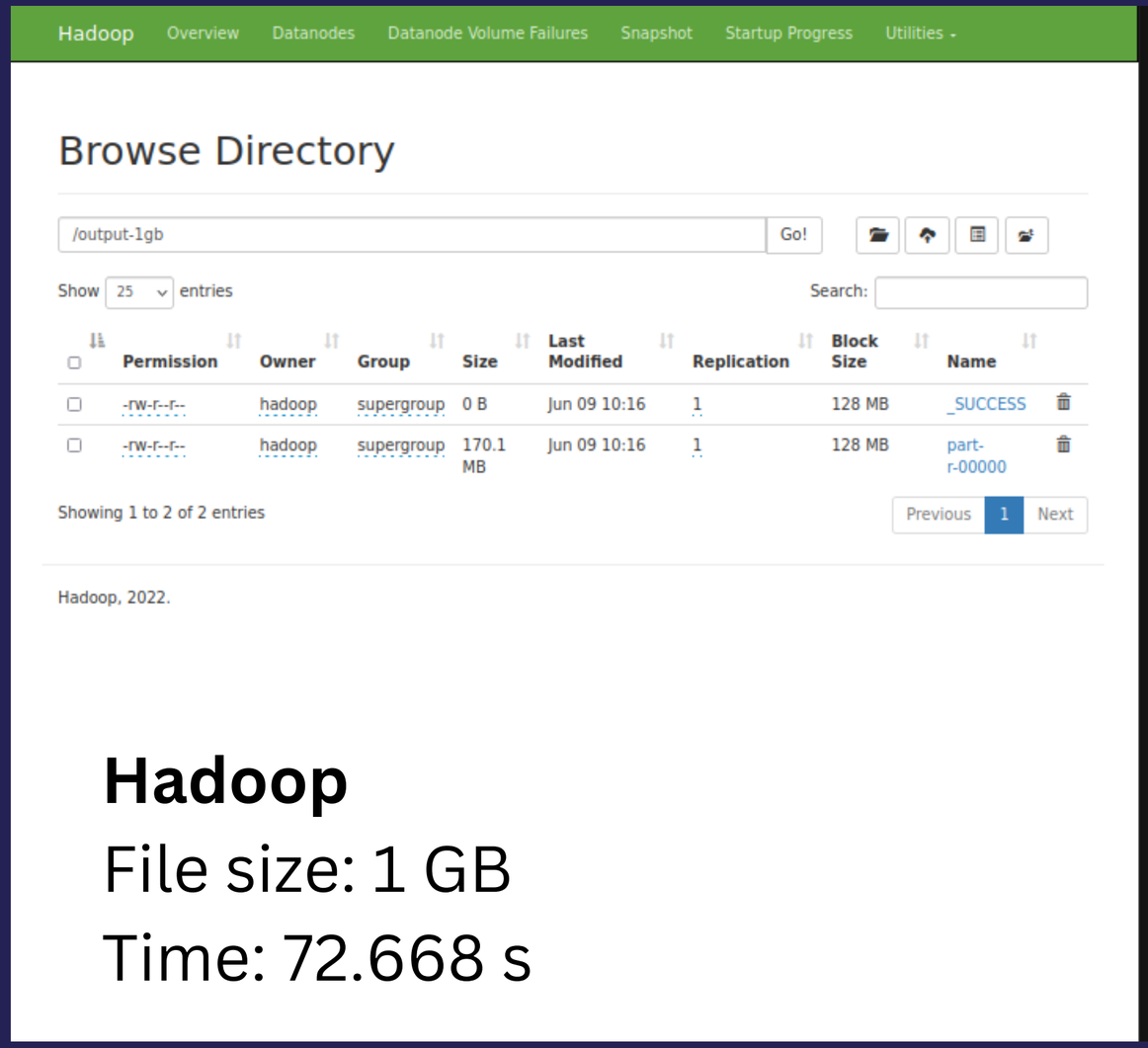


Hadoop

File size: 100 MB

Time: 36.35 s

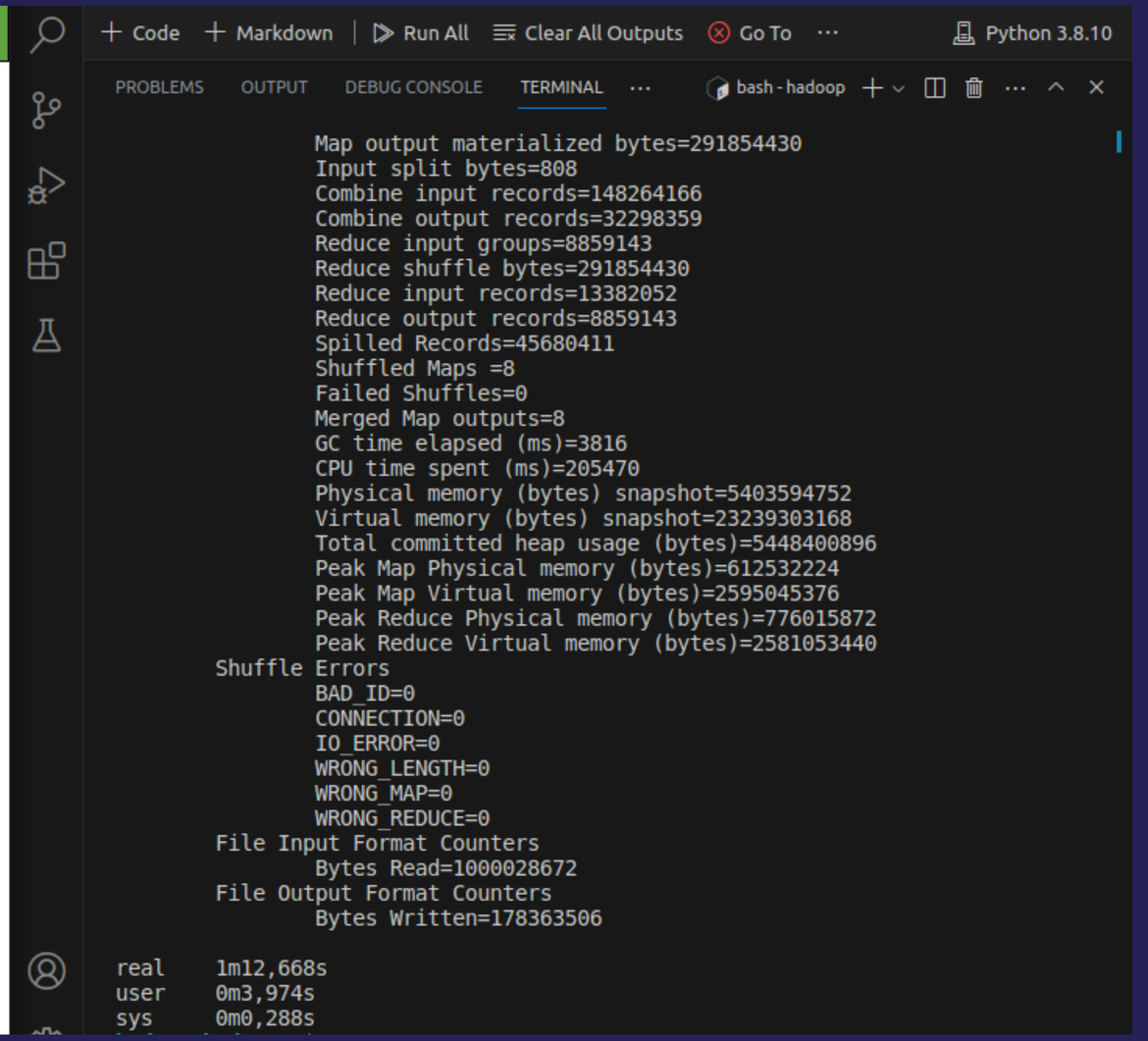


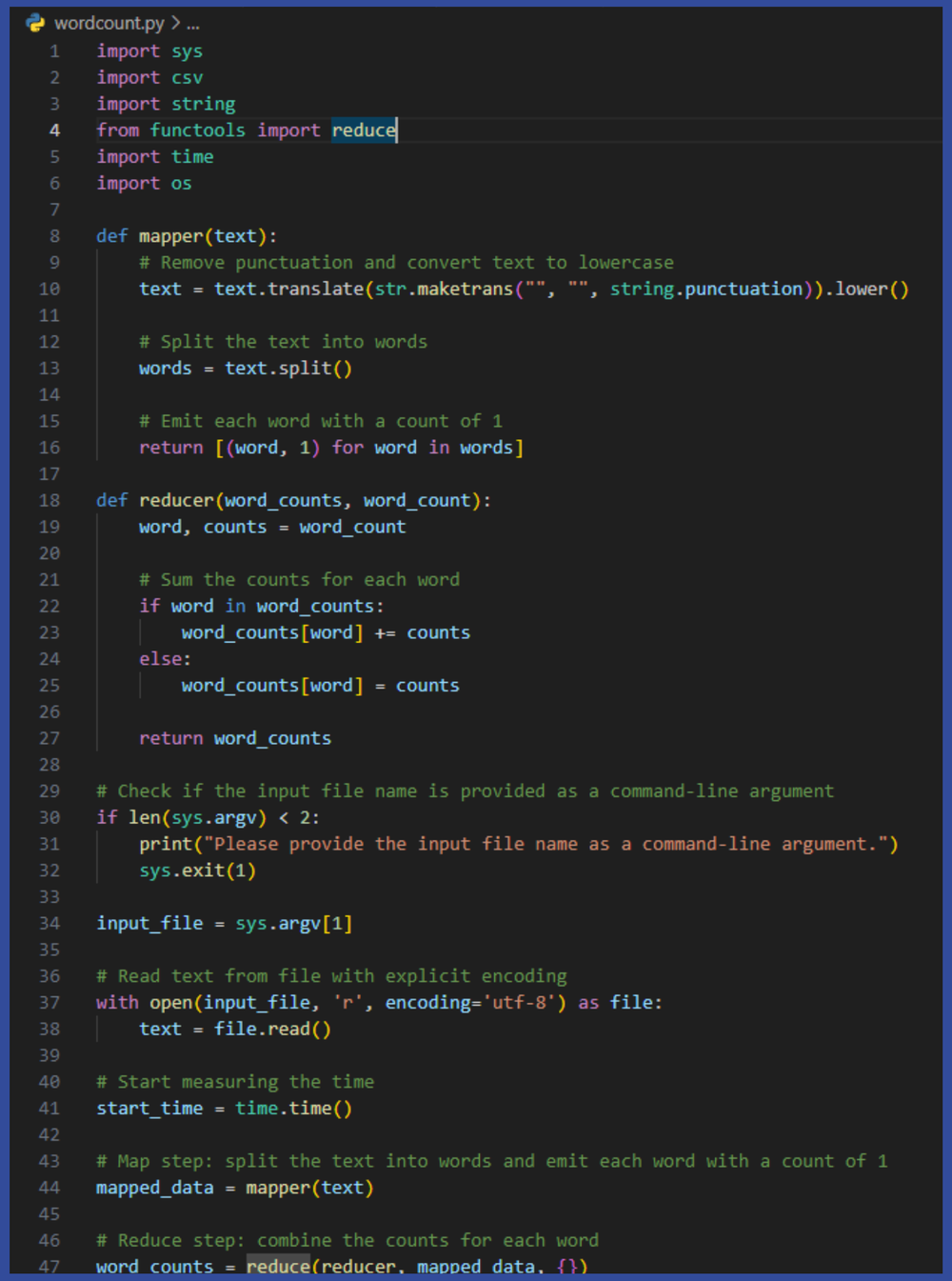


Hadoop

File size: 1 GB

Time: 72.668 s





Features of wordcount.py

- Implements word count algorithm using map() and reduce()
- Read text from file with explicit encoding
- Write the word counts to a CSV file with proper encoding and error handling
- Construct the output file name with the iteration number

References

- https://www.geeksforgeeks.org/python-remove-punctuation-from-string/
- https://www.learnpython.org/en/Map%2C_Filter%2C_Reduce
- https://stackoverflow.com/questions/436220/how-to-determine-the-encoding-of-text
- https://www.knowledgehut.com/blog/programming/sys-argv-python-examples
- https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/Python_FileText.html
- https://www.geeksforgeeks.org/how-to-iterate-over-files-in-directory-using-python/

Python

File size: 100 KB

Time: 0.039 s

```
PROBLEMS 4   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

hadoop@vmbox:~/Downloads/wordcount$ time python3 wordcount.py personae.txt
Word counts saved to: output-1.csv

real    0m0,039s
user    0m0,039s
sys     0m0,000s
hadoop@vmbox:~/Downloads/wordcount$
```

```
# File size: 100 KB
data_1 = pd.read_csv('output-1.csv')
data_1
```

✓ 0.0s

	Word	Count
0	as	183
1	you	543
2	like	83
3	it	226
4	dramatis	1
...	...	...
3275	complexions	1
3276	liked	1
3277	breaths	2
3278	defied	1
3279	curtsy	1

3280 rows × 2 columns

Python

File size: 1 MB

Time: 0.416 s

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

hadoop@vmbox:~/Downloads/wordcount$ time python3 wordcount.py melville.txt
Word counts saved to: output-2.csv

real    0m0,416s
user    0m0,369s
sys     0m0,044s
hadoop@vmbox:~/Downloads/wordcount$
```

```
# File size: 100 KB
data_2 = pd.read_csv('output-2.csv')
data_2
```

✓ 0.0s

	Word	Count
0	the	21630
1	project	102
2	gutenberg	38
3	ebook	28
4	of	10181
...	...	...
27031	obloquy	1
27032	reconcile	1
27033	lecture	1
27034	studydoor	1
27035	thisbe	1

27036 rows × 2 columns

Python

File size: 10 MB

Time: 2.283 s

```
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

hadoop@vmbox:~/Downloads/wordcount$ time python3 wordcount.py shakespeare.txt
Word counts saved to: output-3.csv

real    0m2,283s
user    0m2,187s
sys     0m0,084s
hadoop@vmbox:~/Downloads/wordcount$
```

```
# File size: 10 MB
data_3 = pd.read_csv('output-3.csv')
data_3
```

✓ 0.0s

	Word	Count
0	the	98677
1	project	125
2	gutenberg	41
3	ebook	13
4	of	59991
...	...	...
61601	alleluia	4
61602	chalcedony	1
61603	sardonyx	1
61604	chrysolyte	1
61605	chrysoprasus	1

61606 rows × 2 columns

Python

File size: 100 MB

Time: 17.899 s

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

hadoop@vmbox:~/Downloads/wordcount$ time python3 wordcount.py enwik8.txt
Word counts saved to: output-4.csv

real    0m17,899s
user    0m16,961s
sys     0m0,938s
hadoop@vmbox:~/Downloads/wordcount$
```

```
# File size: 100 MB
data_4 = pd.read_csv('output-4.csv')
data_4
```

✓ 0.5s

	Word	Count
0	mediawiki	25
1	xmlnshttpwwwmediawikiorgxmlexport03	1
2	xmlnsxsihttpwwww3org2001xmlschemainstance	1
3	xsischemalocationhttpwwwmediawikiorgxmlexport03	1
4	httpwwwmediawikiorgxmlexport03xsd	1
...	...	...
760425	nihongokōgo口語	1
760426	kōgo	1
760427	regiontōhoku	1
760428	kyūshū	2
760429	kansaiben	1

760430 rows × 2 columns

Python

File size: 1 GB

Time: ± 318.04 s

Kernel killed after 2 minutes

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

hadoop@vmbox:~/Downloads/wordcount$ time python3 wordcount.py enwik9.txt
Killed

real    2m8,308s
user    1m42,791s
sys     0m16,089s
hadoop@vmbox:~/Downloads/wordcount$
```

```
# File size: 1 GB
data_5 = pd.read_csv('output-5.csv')
data_5
```

✓ 3.0s

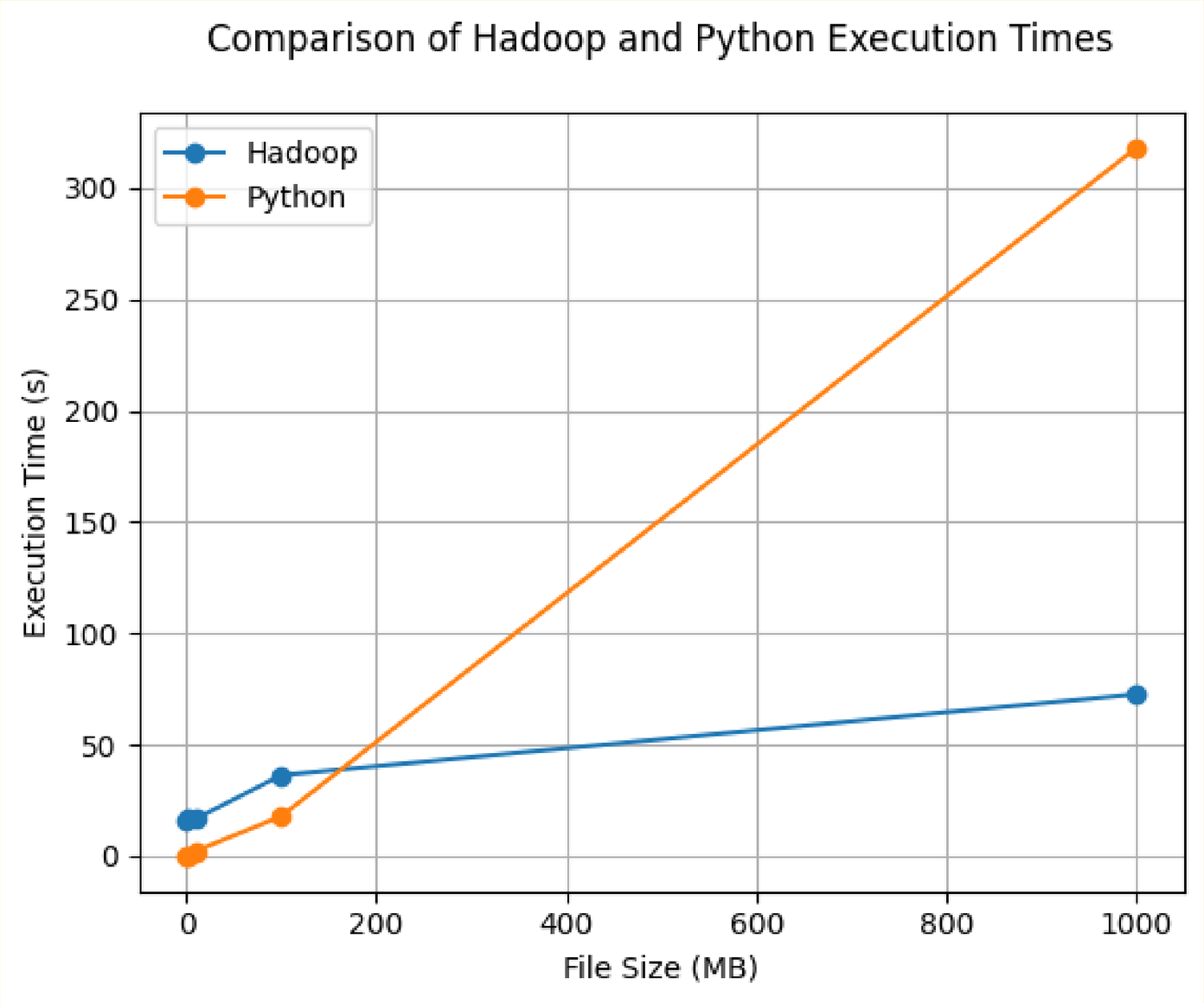
	Word	Count
0	mediawiki	308
1	xmlnshttpwwwmediawikiorgxmlexport03	1
2	xmlnsxsihttpwwww3org2001xmlschemainstance	1
3	xsischemalocationhttpwwwmediawikiorgxmlexport03	1
4	httpwwwmediawikiorgxmlexport03xsd	1
...	...	...
5133222	imagehyperbaricoxygentherapy1personchamberjpgt...	1
5133223	quotmonoplacequot	1
5133224	softsided	1
5133225	httpuhmsorg	1
5133226	uhms	1

5133227 rows × 2 columns

Plot

	100 KB	1 MB	10 MB	100 MB	1 GB
Hadoop	15.82 s	16.53 s	16.74 s	36.35 s	72.66 s
Python	0.039 s	0.416 s	2.283 s	17.899 s	318.04 s

Waktu eksekusi program Word Count lebih cepat dilakukan oleh Python daripada Hadoop untuk file sebesar 100 KB, 1 MB, 10 MB, dan 100 MB. Namun, waktu eksekusi program Python lebih lambat daripada Hadoop untuk file sebesar 1 GB.



Analisis terhadap WordCount

Jika membandingkan waktu eksekusi wordcount dengan python dan menggunakan Hadoop, maka wordcount python lebih cepat karena data yang dihitung disimpan dan dilakukan pada satu mesin atau server tunggal. Sehingga, hanya terjadi sedikit overhead terkait pengaturan dan komunikasi antar node dalam cluster seperti pada Hadoop. Dan untuk ukuran file yang relatif kecil (100 kb, 1 mb, 10 mb) maka wordcount dengan python dapat menyelesaikan tugas lebih cepat karena tidak melibatkan kompleksitas yang signifikan.

Sedangkan, hadoop dirancang untuk memproses dan menganalisis data dalam skala besar dengan membagi tugas pemrosesan ke beberapa node dalam cluster sehingga dapat terjadi overhead yang signifikan walaupun penggunaan hadoop sangat efektif, khususnya pada platform terdistribusi dan data dalam ukuran besar.

Kinerja hadoop menjadi lebih lambat dibandingkan wordcount normal pada skala kecil karena membutuhkan waktu untuk menghubungkan node-node dalam cluster dan komunikasi antar node.

Kesimpulan:

Untuk data yang besar dan membutuhkan pemprosesan dalam skala yang luas, Hadoop menjadi pilihan lebih baik karena lebih optimal. Sedangkan, pada kasus diatas, wordcount normal tidak melibatkan pengaturan dan komunikasi antar node yang kompleks sehingga kinerja lebih cepat dibandingkan menggunakan Hadoop.

THANK YOU