FACULTY OF ENGINEERING

MapReduce and Hadoop

Distributed Systems

Presented by

Name	ID
Aaser Fawzy Zakria Hassan	19015403
Mohamed Ezzat Saad El-Shazly	19016441
Gamal Abdel Hamid Nasef Nowesar	19015550

02/03/2024

Table of Contents

1 Problem Definition

From the problem statement provided, the task involves setting up Hadoop in a Pseudo Distributed Mode, practicing running HDFS commands, creating analytics applications using MapReduce, and executing them using Hadoop.

1. Hadoop Installation and Configuration.
2. HDFS Commands:
    - Understand and execute basic HDFS commands.
    - Create directories and transfer files between local and HDFS.
3. Cloudera Installation (Optional Method):
    - Install Cloudera for an alternative Hadoop setup.
    - Configure Cloudera virtual machine and understand its components.
4. Word Count Application:
    - Develop Word Count application in Java using Eclipse.
    - Add Hadoop libraries, build the application, and create a JAR file.
    - Run the Word Count application as a Hadoop job, check output, and manage results using HDFS commands.

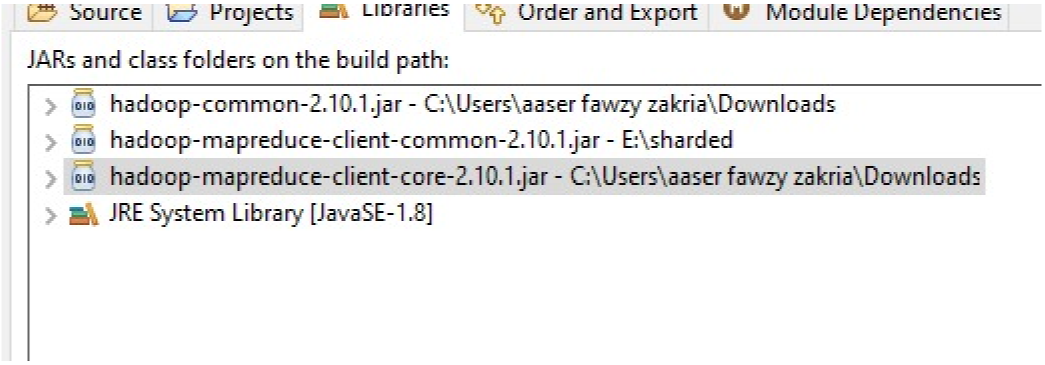
2 Algorithms

1) Hadoop Installation and Configuration.
    - Download Hadoop: Use the provided link to download the Hadoop distribution package.
    - Extract the downloaded file using the command:

    ```
    tar -xzvf hadoop-2.10.1.tar.gz
    ```

    - Setup Hadoop: Follow the steps outlined in the provided documentation to set up Hadoop in Pseudo Distributed Mode.
2) HDFS Commands:
    - Execute Basic Commands: Use the Hadoop command line interface to execute basic HDFS commands such as listing files, creating directories, and copying files.
    - Transfer Files: Employ Hadoop commands to create directories and transfer files between the local file system and HDFS.
3) Cloudera Installation (Optional Method): skipped
4) Word Count Application:
    - Develop the Word Count application using Java within the Eclipse IDE.
    - Add the necessary Hadoop libraries to the project's build path to enable interaction with the Hadoop framework.

- Implement the mapper and reducer classes to perform word counting using the MapReduce programming model.
- Create a JAR file containing the application and its dependencies using the hadoop-common-2.10.1 ,hadoop-mapreduce-client-common-2.10.1 and hadoop-mapreduce-client-core-2.10.1 jars.



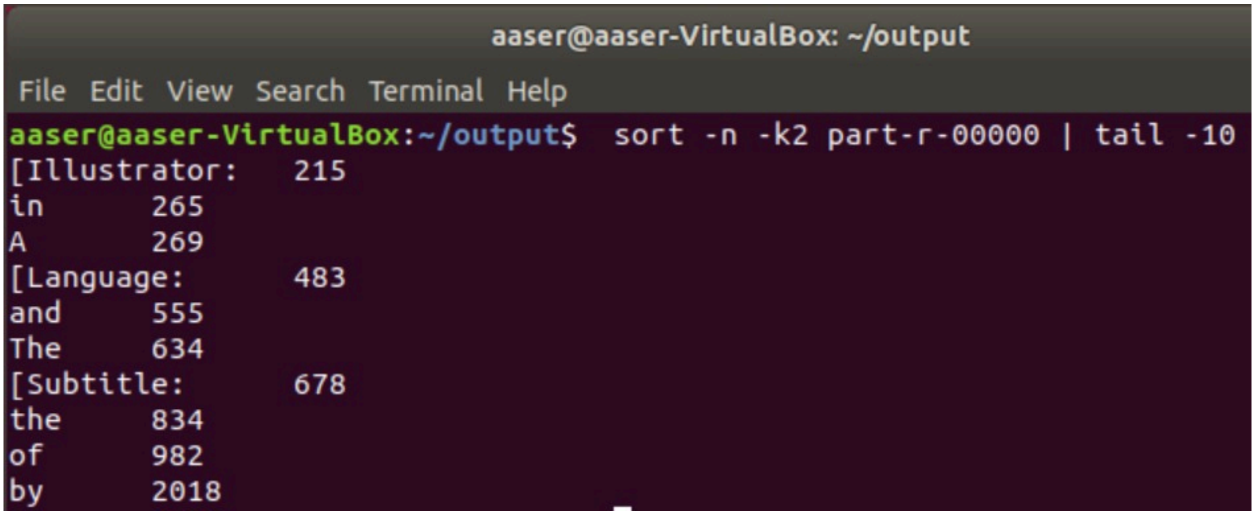
JARs and class folders on the build path:

```
hadoop-common-2.10.1.jar - C:\Users\aaser fawzy zakria\Downloads
hadoop-mapreduce-client-common-2.10.1.jar - E:\sharded
hadoop-mapreduce-client-core-2.10.1.jar - C:\Users\aaser fawzy zakria\Downloads
JRE System Library [JavaSE-1.8]
```

- Execute the Word Count application as a Hadoop job using the Hadoop command line:

```
bin/hadoop jar wc.jar WordCount /user/aaser/input /user/aaser/output
```

- Check the output files generated by the Word Count application in the specified output directory using HDFS commands. Manage the results as necessary.


aaser@aaser-VirtualBox: ~/output
File Edit View Search Terminal Help

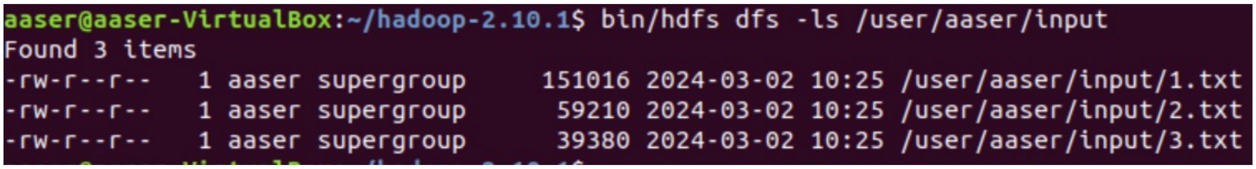
```
aaser@aaser-VirtualBox:~/output$  sort -n -k2 part-r-00000 | tail -10
[Illustrator:   215
in      265
A       269
[Language:      483
and     555
The     634
[Subtitle:      678
the     834
of      982
by      2018
```

3 Implementation

- Environment:
  - Local machine setup using a personal laptop in a virtualized environment using VirtualBox.
  - Network Type: We use SSH for local machine connection through localhost setup does not involve network communication between nodes. It operates as a Pseudo Distributed mode so the communication is between processes each acting as a node.
- Machine Specifications (virtualbox os):
  - Operating System: Ubuntu 18.04
  - RAM: 4 GB.
  - Processor: Multi-core processor (Intel Core i5).
- Test Data (Dataset):
  - Gutenberg Project Texts, as it offers diverse text formats and sizes, facilitating comprehensive testing of the Word Count application.
- Number of Runs:
  - Multiple runs on the dataset: Executed the algorithm multiple times to assess consistency, performance under different conditions, and identify potential issues.

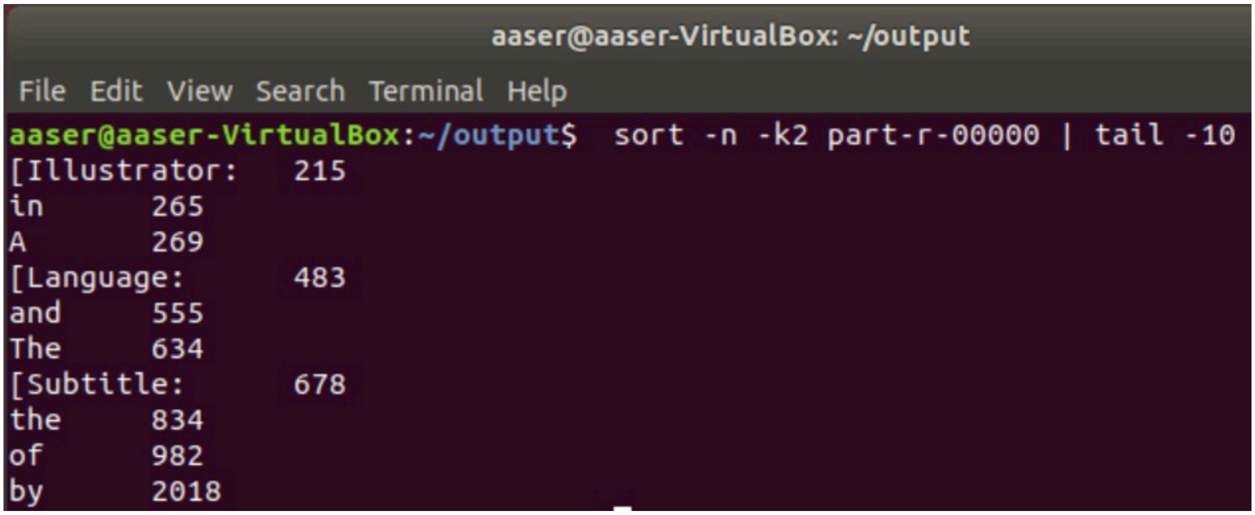
4 Results

list the input files:



```
aaser@aaser-VirtualBox:~/hadoop-2.10.1$ bin/hdfs dfs -ls /user/aaser/input
Found 3 items
-rw-r--r--   1 aaser supergroup     151016 2024-03-02 10:25 /user/aaser/input/1.txt
-rw-r--r--   1 aaser supergroup      59210 2024-03-02 10:25 /user/aaser/input/2.txt
-rw-r--r--   1 aaser supergroup      39380 2024-03-02 10:25 /user/aaser/input/3.txt
```

the final output:



aaser@aaser-VirtualBox: ~/output
File Edit View Search Terminal Help

```
aaser@aaser-VirtualBox:~/output$  sort -n -k2 part-r-00000 | tail -10
[Illustrator:   215
in      265
A       269
[Language:      483
and     555
The     634
[Subtitle:      678
the     834
of      982
by      2018
```

5 Conclusion

- The assignment enhanced practical understanding of Hadoop installation, HDFS management, and MapReduce programming model. It provided hands-on experience in setting up a Hadoop environment and running analytics applications.
- Understanding how to work with Hadoop and MapReduce is highly relevant in today's data-driven world. These skills are essential for processing and analyzing large-scale datasets efficiently.