

EXPERIMENT NO: 3

Student Name: Ansari Mohammed Bilal

PRN: 211207

Course Name: Big Data Analytics

Course Code: CSC702

Dept. name of organization: Computer

Name of organization: M. H. Saboo Siddik College of Engineering

AIM: Implement the Word Count program using Hadoop MapReduce

Keywords— Hadoop, Map Reduce, Word Count, JAVA, Cloudera

Abstract

The Word Count Problem is a fundamental task in the realm of big data processing and serves as an introductory exercise for understanding Hadoop's capabilities. In Cloudera's Hadoop environment, the Word Count Problem involves counting the frequency of each word in a large dataset. This problem is typically approached using Hadoop's MapReduce framework, which divides the data processing task into two phases: Map and Reduce. In the Map phase, data is split into manageable chunks and processed by mapper functions to produce key-value pairs, where the key is the word and the value is its count. The Reduce phase then aggregates these key-value pairs, summing up the counts for each word to produce the final output. Leveraging Cloudera's distribution of Hadoop, which includes tools like Apache Hive and Apache Pig, enhances the efficiency and scalability of this process, allowing for the handling of vast amounts of data across distributed clusters. The Word Count Problem not only demonstrates the basic principles of Hadoop's distributed computing but also illustrates the power of Cloudera's ecosystem in managing and analyzing big data effectively.

I. Introduction

The Word Count Problem is a classic example used to illustrate the principles of distributed computing and data processing with Hadoop. As data volumes continue to grow exponentially, traditional methods of data analysis become insufficient, necessitating the use of scalable and efficient tools. Hadoop, an open-source framework designed for distributed storage and processing, provides a robust solution to this challenge. Within Hadoop, the MapReduce programming model is instrumental in performing large-scale data processing tasks by dividing the work into manageable units. Cloudera, a leading provider of Hadoop-based solutions, offers a comprehensive platform that enhances the Hadoop ecosystem with additional tools and optimizations.

In the context of the Word Count Problem, Cloudera's distribution facilitates efficient processing by leveraging its advanced Hadoop ecosystem, which includes components like Apache HDFS for distributed storage, and Apache YARN for resource management. By distributing the word counting task across a cluster of machines, Hadoop and Cloudera's technologies enable the processing of vast datasets with high performance and fault tolerance. This introduction to the Word Count Problem within the Cloudera environment sets the stage for exploring how

Hadoop's MapReduce model operates in practice and highlights the advantages of using Cloudera's platform for handling large-scale data processing tasks.

A. Input file

The image consists of two screenshots of a Cloudera Quickstart VM terminal window. The top screenshot shows the initial state where the user lists files in the current directory and then runs a command to create a directory. The bottom screenshot shows the user running a Hadoop command to upload a file to the HDFS filesystem, followed by a Java exception stack trace.

```
cloudera@quickstart:~$ ls
cloudera-manager  cp-fr-local.txt  Desktop  Downloads  enterprise-deployment.json  kerberos  lib  Music  Pictures  Public  Videos  workspace
cn api.py         desktop          Documents  eclipse     express-deployment.json  Kulsom  local_exp2  parcels  ProcessFile.txt  Templates  WordCount.jar

cloudera@quickstart:~$ cat > /home/cloudera/ProcessFile.txt
0oooooooooooooooooehh

Tipe, tipe, zangalewa. World cup! World cup!
Tipe, tipe, zangalewa. World cup! World cup!

Tipe, tipe, zangalewa. World cup! World cup!
Tipe, tipe, zangalewa. World cup! World cup!

You're a good soldier
Choosing your battles
Pick yourself up
And dust yourself off
Get back in the saddle

You're on the front line
Everyone's watching
You know it's serious
We're getting closer
This isn't over

The pressure's on
You feel it
But you got it all
Believe it

When you fall get up, oh oh
If you fall get up, eh eh
Tsamina mina zangalewa
'cause this is Africa

Tsamina mina, eh eh
Waka waka, eh eh
Tsamina mina zangalewa
This time for Africa

Listen to your God

cloudera@quickstart:~$ hadoop fs -put /home/cloudera/ProcessFile.txt /InputFolder
put: '/InputFolder/ProcessFile.txt': File exists
cloudera@quickstart:~$ hadoop fs -put /home/cloudera/ProcessFile2.txt /InputFolder
24/07/31 03:07:36 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
cloudera@quickstart:~$ hadoop fs -cat /InputFolder/ProcessFile2.txt
0oooooooooooooooooehh

Tipe, tipe, zangalewa. World cup! World cup!
Tipe, tipe, zangalewa. World cup! World cup!

Tipe, tipe, zangalewa. World cup! World cup!
Tipe, tipe, zangalewa. World cup! World cup!

You're a good soldier
Choosing your battles
Pick yourself up
And dust yourself off
Get back in the saddle

You're on the front line
Everyone's watching
You know it's serious
We're getting closer
This isn't over

The pressure's on
You feel it
But you got it all
Believe it

When you fall get up, oh oh
If you fall get up, eh eh
Tsamina mina zangalewa
'cause this is Africa

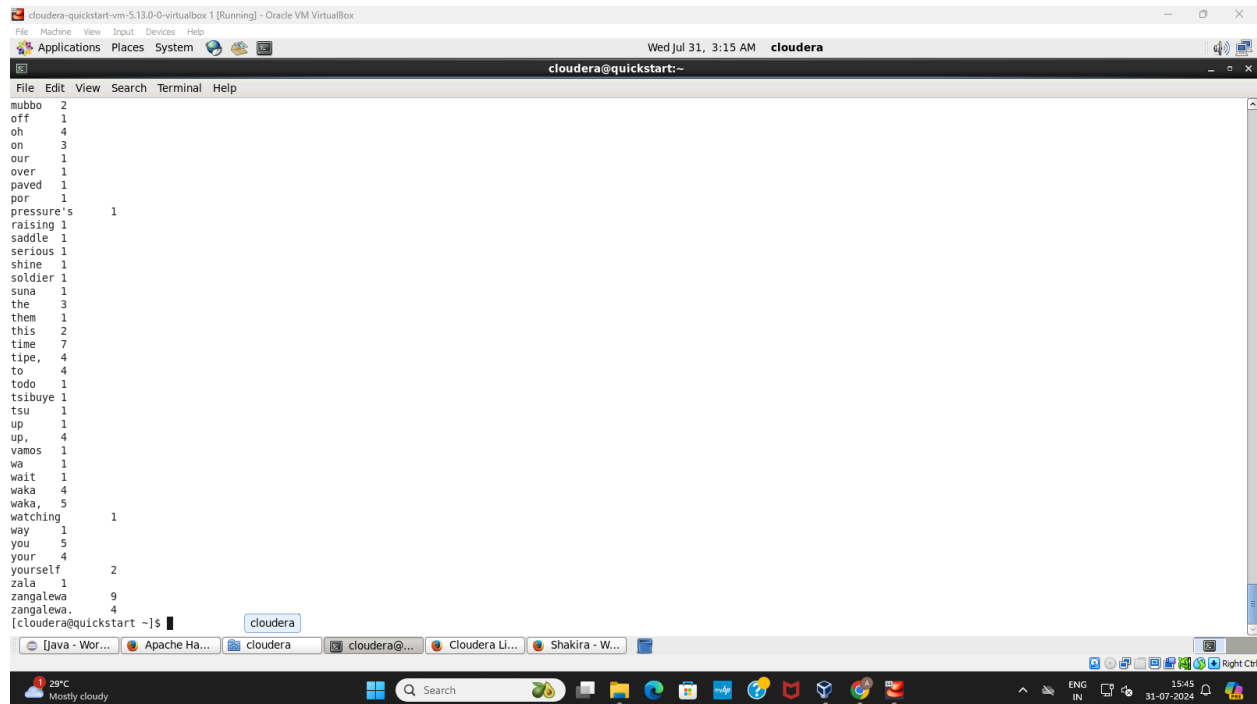
Tsamina mina, eh eh
Waka waka, eh eh
Tsamina mina zangalewa
This time for Africa

Listen to your God
```

B. Output

```
cloudera-quickstart-vm-5.13.0-0-virtualbox 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -ls /ResultFolder
Found 1 items
drwxr-xr-x - cloudera supergroup 0 2024-07-31 03:13 /ResultFolder/Bilal
[cloudera@quickstart ~]$ hadoop fs -cat /ResultFolder/Bilal/part-r-00000
'Cause 1
'cause 1
A 1
Africa 8
Africa... 1
Ane 1
Anawa 6
And 1
Asi 1
Bathi 1
Believe 2
But 1
Choosing 1
Django 4
Don't 1
East 1
Everyone's 1
Get 1
Go 1
God 1
I 1
If 2
Listen 1
No 1
Oooooooooooooooooooooohh 1
People 1
Pick 1
Tendency 1
The 1
Their 1
This 9
Tipe, 4
Today's 1
Tsamina 17
Waka 5
We're 2
```

```
cloudera-quickstart-vm-5.13.0-0-virtualbox 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
[Zolani:] 1
a 13
all 2
and 1
are 1
back 1
battles 1
biggi 4
closer 1
cup! 8
day 1
down 2
dust 1
eh 36
expectations 1
fall 2
feed 1
feel 2
for 6
from 1
front 1
get 6
getting 1
good 1
got 1
hesitation 1
in 2
is 4
isn't 1
it 5
it's 1
know 1
la 1
line 2
ma 2
mejole 2
mina 10
mina, 7
moment 1
```



The screenshot shows a terminal window titled "cloudera@quickstart:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the output of a word count command, listing words and their frequencies. The words and counts are: mubbo 2, off 1, oh 4, on 3, our 1, over 1, paved 1, por 1, pressure's 1, raising 1, saddle 1, serious 1, shine 1, soldier 1, suna 1, the 3, them 1, this 2, time 7, tipe, 4, to 4, todo 1, tsibuye 1, tsu 1, up 1, up, 4, vamos 1, wa 1, wait 1, waka 4, waka, 5, watching 1, way 1, you 5, your 4, yourself 2, zala 1, zangalewa 9, zangalewa, 4. The prompt "[cloudera@quickstart ~]\$" is visible at the bottom of the terminal. The window is part of a larger desktop environment with a taskbar at the bottom showing various application icons and system status information (23°C, Mostly cloudy, 15:45, 31-07-2024).

```
mubbo 2
off 1
oh 4
on 3
our 1
over 1
paved 1
por 1
pressure's 1
raising 1
saddle 1
serious 1
shine 1
soldier 1
suna 1
the 3
them 1
this 2
time 7
tipe, 4
to 4
todo 1
tsibuye 1
tsu 1
up 1
up, 4
vamos 1
wa 1
wait 1
waka 4
waka, 5
watching 1
way 1
you 5
your 4
yourself 2
zala 1
zangalewa 9
zangalewa, 4
[cloudera@quickstart ~]$
```

II. Conclusion

The conclusion of the MapReduce process in Hadoop underscores its pivotal role in enabling large-scale data processing by efficiently distributing and parallelizing tasks across a cluster of nodes. By breaking down a job into smaller map and reduce tasks, MapReduce allows for the processing of vast datasets in a fault-tolerant and scalable manner. The map phase is responsible for filtering and sorting data, while the reduce phase aggregates and consolidates the results, ultimately producing the final output. This framework's ability to handle massive volumes of data with reliability and speed has made it a cornerstone technology in the realm of big data, powering everything from search engines to data analytics platforms.

REFERENCES

- [1] Apache Hadoop, "MapReduce Tutorial," [Online]. Available: <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>. [Accessed: 10-Aug-2024].
- [2] Exp 5 - Map Reducer - WordCount.docx, Google Classroom, 2024. [Accessed: 30-July-2024].