Expt-2

Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

# AIM:

To run a basic Word Count MapReduce program using Hadoop.

# PROCEDURE:

1. **Create Data File**:

nano word\_count\_data.txt

# Example content for word\_count\_data.txt:

Hadoop is a framework that allows for distributed processing of large data sets.

1. **Mapper Program (mapper.py)**: import sys

for line in sys.stdin: line = line.strip() words = line.split() for word in words:

print(f'{word}\t1')

# Reducer Program (reducer.py):

import sys current\_word = None current\_count = 0 word = None

for line in sys.stdin: line = line.strip()

word, count = line.split('\t', 1)

try:

count = int(count) except ValueError:

continue

if current\_word == word: current\_count += count

else:

if current\_word: print(f'{current\_word}\t{current\_count}')

current\_count = count current\_word = word

if current\_word == word: print(f'{current\_word}\t{current\_count}')

# Set Hadoop Environment:

hdfs dfs -mkdir /word\_count\_input

hdfs dfs -copyFromLocal word\_count\_data.txt /word\_count\_input

# Run Word Count Program:

hadoop jar $HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-\*.jar \

-input /word\_count\_input/word\_count\_data.txt \

-output /word\_count\_output \

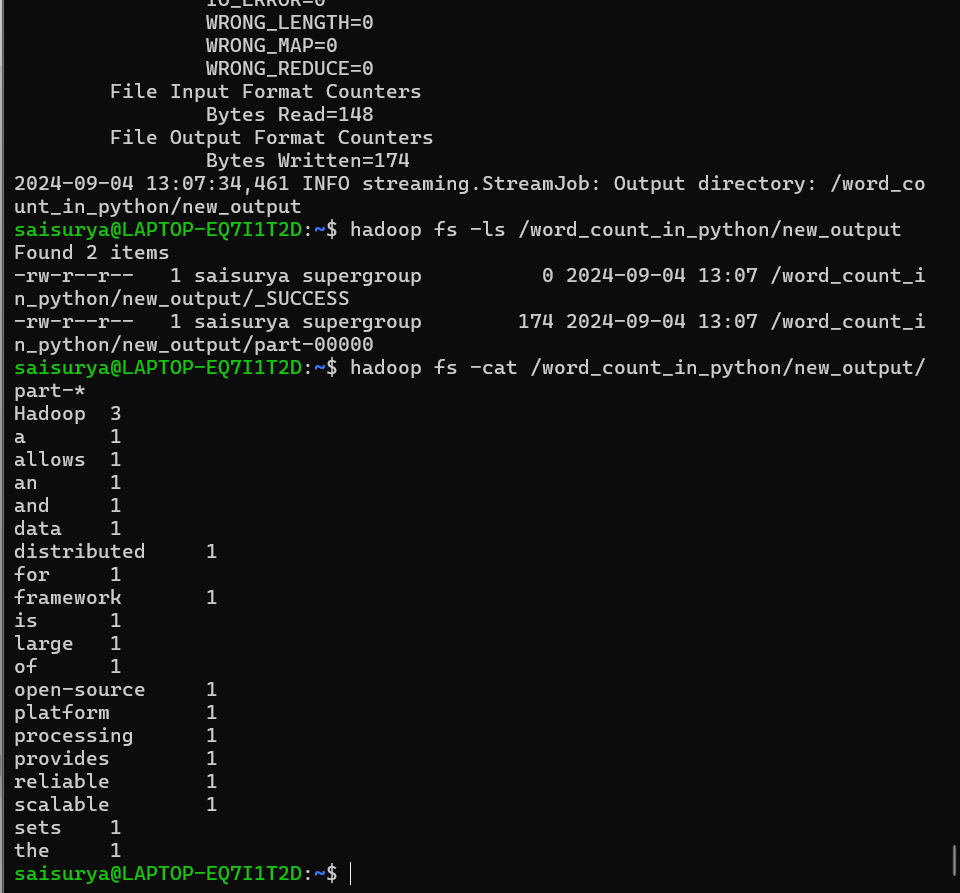
-mapper mapper.py \

-reducer reducer.py

# Check Output:

hdfs dfs -cat /word\_count\_output/part-00000

# OUTPUT:



**RESULT:**

Thus, the program for basic Word Count Map Reduce has been executed successfully.