**MIS 381N Information Management**

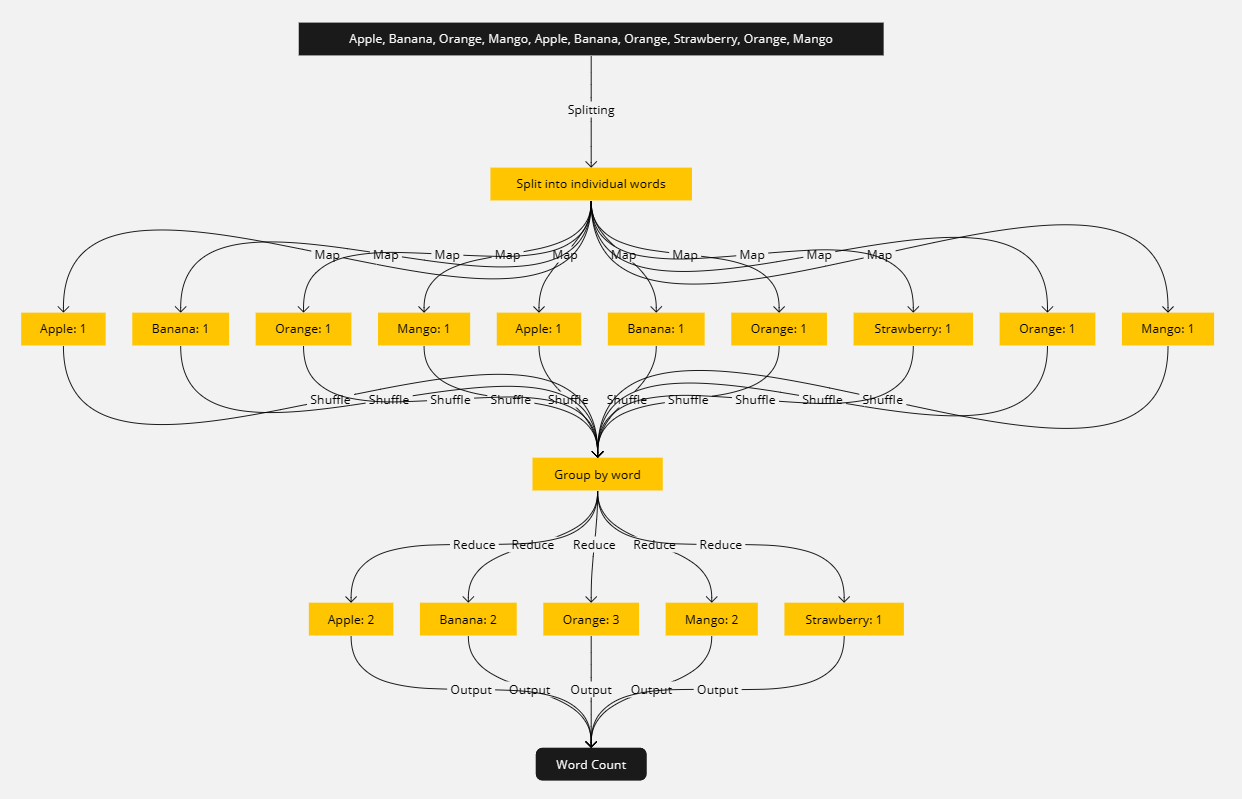
**Assignment 4: MapReduce and Hadoop**

**-Group Members**

Karthick Vel Kathirvel (kk37347)

Jahnavi Angati (ja54632)

1. **Create a MapReduce diagram to illustrate the process of counting word frequencies using MapReduce.  
   Note: There is no right or wrong way in the “splitting” step as long as the entire MapReduce process is done correctly.  
   Input:  
   Apple, Banana, Orange, Mango, Apple, Banana, Orange, Strawberry, Orange, Mango  
   Perform a word count on the above using MapReduce.**



1. **Install Hadoop (or Hadoop Streaming interface for python users) and  
   familiarize yourself with the environment. There are plenty of useful documents online.  
   Paste a screenshot of your Hadoop interface.**

A screenshot of a computer

Description automatically generated

1. **Implement a Python code to count word frequencies and use the concepts from MapReduce (Even if you are not using Hadoop, please make sure that you are using concepts from MapReduce in your code). If you do use Hadoop, you will have to install the Hadoop streaming API. Submit code and output.**

**Files used:**

* input.txt
* mapper.py
* reducer.py

**Output:**

* \_SUCCESS
* part-00000

**input.txt:**

Two roads diverged in a yellow wood,

And sorry I could not travel both

And be one traveler, long I stood

And looked down one as far as I could

To where it bent in the undergrowth;

Then took the other, as just as fair,

And having perhaps the better claim,

Because it was grassy and wanted wear;

Though as for that the passing there

Had worn them really about the same,

And both that morning equally lay

In leaves no step had trodden black.

Oh, I kept the first for another day!

Yet knowing how way leads on to way,

I doubted if I should ever come back.

I shall be telling this with a sigh

Somewhere ages and ages hence:

Two roads diverged in a wood, and I—

I took the one less traveled by,

And that has made all the difference.

**Code used in terminal:**

hadoop jar $HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-\* -files /home/kkarthick12/mapper.py,/home/kkarthick12/reducer.py -mapper /home/kkarthick12/mapper.py -reducer /home/kkarthick12/reducer.py -input /home/kkarthick12/input.txt -output /home/kkarthick12/output

**mapper.py:**

#!/usr/bin/env python3

import sys  
import re

def mapper():

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# split the line into words

words = re.findall(r'\w+', line)

# output each word as a key-value pair

for word in words:

print(f'{word}\t1')

if \_\_name\_\_ == '\_\_main\_\_':

mapper()

**reducer.py:**

#!/usr/bin/env python3

import sys

def reducer():

current\_word = None

current\_count = 0

for line in sys.stdin:

# parse the input from mapper.py

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

# convert count (currently a string) to int

try:

count = int(count)

except ValueError:

continue

# this IF-switch only works because Hadoop sorts map output

if current\_word == word:

current\_count += count

else:

if current\_word:

# write result to STDOUT

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

current\_word = word

current\_count = count

if current\_word == word:

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

if \_\_name\_\_ == '\_\_main\_\_':

reducer()

**Output:**

**part-00000**

|  |  |
| --- | --- |
| **Word** | **Count** |
| And | 6 |
| Because | 1 |
| Had | 1 |
| I | 9 |
| In | 1 |
| Oh | 1 |
| Somewhere | 1 |
| Then | 1 |
| Though | 1 |
| To | 1 |
| Two | 2 |
| Yet | 1 |
| a | 3 |
| about | 1 |
| ages | 2 |
| all | 1 |
| and | 3 |
| another | 1 |
| as | 5 |
| back | 1 |
| be | 2 |
| bent | 1 |
| better | 1 |
| black | 1 |
| both | 2 |
| by | 1 |
| claim | 1 |
| come | 1 |
| could | 2 |
| day | 1 |
| difference | 1 |
| diverged | 2 |
| doubted | 1 |
| down | 1 |
| equally | 1 |
| ever | 1 |
| fair | 1 |
| far | 1 |
| first | 1 |
| for | 2 |
| grassy | 1 |
| had | 1 |
| has | 1 |
| having | 1 |
| hence | 1 |
| how | 1 |
| if | 1 |
| in | 3 |
| it | 2 |
| just | 1 |
| kept | 1 |
| knowing | 1 |
| lay | 1 |
| leads | 1 |
| leaves | 1 |
| less | 1 |
| long | 1 |
| looked | 1 |
| made | 1 |
| morning | 1 |
| no | 1 |
| not | 1 |
| on | 1 |
| one | 3 |
| other | 1 |
| passing | 1 |
| perhaps | 1 |
| really | 1 |
| roads | 2 |
| same | 1 |
| shall | 1 |
| should | 1 |
| sigh | 1 |
| sorry | 1 |
| step | 1 |
| stood | 1 |
| telling | 1 |
| that | 3 |
| the | 8 |
| them | 1 |
| there | 1 |
| this | 1 |
| to | 1 |
| took | 2 |
| travel | 1 |
| traveled | 1 |
| traveler | 1 |
| trodden | 1 |
| undergrowth | 1 |
| wanted | 1 |
| was | 1 |
| way | 2 |
| wear | 1 |
| where | 1 |
| with | 1 |
| wood | 2 |
| worn | 1 |
| yellow | 1 |