TP1 Hadoop

EXO2

Step 1: Write the MapReduce Code

Create two Python scripts:

- `mapper.py` (extract words from purchases.txt)

```
#!/usr/bin/env python3
import sys

for line in sys.stdin:
    words = line.strip().split()
    for word in words:
        print("{}\t1".format(word))
```

- `reducer.py` (count occurrences of each word)

```
#!/usr/bin/env python3import sys
current_word = None
current_count = 0
for line in sys.stdin:
    word, count = line.strip().split('\t')
    count = int(count)
    if word == current_word:
        current_count += count
    else:
        if current_word:
          print("{}\t{}".format(current_word, current_count))
        current_word = word
        current_count = count
```

```
if current_word:
    print("{}\t{}".format(current_word, current_count))
```

write in git bash(if u are in windows) this command so that the file be executable:

```
chmod 777 mapper.py reducer.py
```

Step 2: Set Up and Start Hadoop Cluster

1. Start Docker containers (master + 2 slaves):

```
docker start hadoop-master hadoop-slave1 hadoop-slave2
```

2. Access the master node:

```
docker exec -it hadoop-master bash
```

3. Start Hadoop inside the container:

```
./start-hadoop.sh
```

Step 3: Prepare HDFS (Hadoop Distributed File System)

1. Create an input directory in HDFS:

```
hadoop fs -mkdir -p input
```

2. Copy `purchases.txt` into the Hadoop container:

```
docker cp purchases.txt hadoop-master:/root/
```

3. Move `purchases.txt` into HDFS:

```
hadoop fs -put purchases.txt input
```

Step 4: Copy Mapper and Reducer to Hadoop Master

1. Copy the scripts to the container:

```
docker cp mapper.py hadoop-master:/root/
docker cp reducer.py hadoop-master:/root/
```

Step 5: Download Hadoop Streaming JAR

1. Download `hadoop-streaming-2.7.3.jar` from here and copy it to the container:

```
docker cp hadoop-streaming-2.7.3.jar hadoop-master:/root/
```

Step 6: Execute the MapReduce Job

Now, inside the Hadoop master container, run this command:

```
hadoop jar hadoop-streaming-2.7.2.jar \
    -input input/purchases.txt \
    -output output \
    -mapper /root/"python3 mapper.py" \
    -reducer /root/"python3 reducer.py" \
    -file /root/mapper.py \
    -file /root/reducer.py
```

NOTE:

- In hadoop , python version is 3.5 so verify that the code is in the same version
- If u have any other problem write this command in git bash : `dos2unix mapper.py reducer.py`
- When u run the command hadoop jar ... if u get an error about the output file just delete it `hadoop fs -rm -r output` and rerun again.

Step 7: View the Output

1. View the result:

```
hadoop fs -cat output/part-00000
```

EXO3

Same steps as exo2!

- **mapper_3.py:**

```
#!/usr/bin/env python3import sys
for line in sys.stdin:
fields = line.strip().split('\t')
if len(fields) != 6:
    continue
try:
    store = fields[2]
    cost = float(fields[4])
    print("{}\t{}".format(store, cost))
except ValueError:
    continue
```

- **reducer_3.py:**

```
#!/usr/bin/env python3import sys
current_store = Nonetotal_sales = 0
for line in sys.stdin:
    store, cost = line.strip().split('\t')
    cost = float(cost)
    if store == current_store:
        total_sales += cost
    else:
        if current_store:
            print("{}\t{:.2f}".format(current_store, total_sales))
        current_store = store
        total_sales = cost
```

```
if current_store:
    print("{}\t{:.2f}".format(current_store, total_sales))
```