Student Name: Mai Ngo

Course Name and Number: CSC 555 Big Data Mining

Assignment 2

Date: 10/10/2023

**Part 1**

**1) Exercise 2.2.1 : Suppose we execute the word-count MapReduce program described in this section on a large repository such as a copy of the Web. We shall use 100 Map tasks and some number of Reduce tasks.**

**(a) Suppose we do not use a combiner at the Map tasks. Do you expect there to be significant skew in the times taken by the various reducers to process their value list? Why or why not?**

Since a combiner is not being applied here, we can expect there to be a significant skew in the times taken by the various reducers to process their value list when running the word-count MapReduce program on a large repository like the Web. In MapReduce framework, the dictionary outputs from Map tasks will then be distributed to Reduce tasks using the dictionary key. Let say both second Map task and eighth Map task could have the same dictionaries for key ‘the’ – a common word, could be different occurrence counts. Without the combiner, one of the Reduce tasks will accumulate all dictionaries with the same key ‘the’. Thus, resulting much more processing volume then other Reduce tasks, and vice versa for less common words. Ultimately this causes uneven data distribution, different in Reduce tasks execution time thus, significant time skewness. In addition, without a combiner, network traffic may get really bad since data is not being aggregated properly and distributed all over the place.

**2) Exercise 2.3.5 : The relational-algebra operation R(A, B) ⊲⊳ B< C S(C, D). Produces all tuples (a, b, c, d) such that tuple (a, b) is in relation R, tuple (c, d) is in S, and b < c. Give a MapReduce implementation of this operation, assuming R and S are sets.**

Relation R - (a, b): (1, 2) and (7, 5)

Relation S - (c, d): (3, 10) and (7, 8) and (7, 9)

Expected outputs: (1 2 3 10), (1 2 7 8), (7 5 7 8)

**Map()**

* Input <R, tuple(a, b)> #For each tuple (a, b) in relation R.
* Input <S, tuple(c, d)> #For each tuple (c, d) in relation S.

#Set b and c as keys.

* Parses each block and emits <b, (R, a)> for R tuples, and <c, (S, d)> for S tuples
* b, (R, a1) b, (R, a2)

**Reduce()**

#Group tuple with the same key value of b and c, respectively together.

* Input: <b, [(R,a1), (R,a2)]>, <c, [(R,d1), (R,d2)]>
* **Relation R - 2, [(R, 1), (R, 3)]**
* **Relation S - 3, [(S, 10)] 7, [(S, 8), (S, 9)]**
* Does: Check for key b < c using for loop; construct all pairs
* Output: <a1, b, c, d1> <a1, b, c, d2>

**3) Consider a Hadoop job that processes an input data file of size equal to 165 disk blocks (165 different blocks, you can assume that HDFS replication factor is set to 1). The mapper in this job requires 1 minute to read and process a single block of data. For the purposes of this assignment, you can assume that the reduce part of this job takes zero time. You can also refer to the supplemental example on how to make this estimate.**

You can ignore the network transfer costs and other potential overheads as well as the  
possibility of node failure. State any assumptions you make. **a) Approximately how long will it take to process the file if you only had one Hadoop worker node? You can assume that that only one mapper is created on every node.**

Mapper: Each block takes 1 minute to process.

Reducer: 0 minute. System tasks: 0 minute. Number of nodes: 1

Total time = 1 \* 165 = **165 minutes.**

**b) 10 Hadoop worker nodes?**

Total number of execution rounds: 165 / 10 = 16.5. Therefore, 1 \* 17 = **17 minutes.**

Last round only the first 5 nodes process.

**c) 30 Hadoop worker nodes?**  
Total number of execution rounds: 165 / 30 = 5.5. Therefore, 1 \* 6 = **6 minutes**.

Similar, last round only the first 5 nodes process.

**d) 100 Hadoop worker nodes?**  
Total number of execution rounds: 165 / 100 = 1.65. Therefore, 1 \* 2 = **2 minutes**.

Last round (2nd round) only the first 65 nodes process.

**e) Now suppose you were told that the replication factor has been changed to 3. That is,  
each block is stored in triplicate, but file size is still 165 blocks. Which of the answers (if any) in a)-d) above will have to change?**

With replication of 3 for each block. Total number of blocks is 165 \* 3 **= 495 blocks.**

The running time maybe increasing somewhat due to data replication. However, consider we are ignoring the network transfer costs and other potential overheads, which we can increasing the speed (bytes/sec). Therefore, total running time will be still the same.

**Part 2 Note:** Code comment in html file. **| HDFS folder:** hadoop fs -mkdir /data

Query 1)

**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/supplier.tbl -output /data/output1 -mapper query1\_mapper.py -reducer query1\_reducer.py -file query1\_mapper.py -file query1\_reducer.py

**Mapper:**

A screenshot of a computer code

Description automatically generated

**Reducer:**

**A screenshot of a computer

Description automatically generated**

**Output:**

A screenshot of a computer

Description automatically generated

Query 2)

**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl -output /data/output2 -mapper query2\_mapper.py -reducer query2\_reducer.py -file query2\_mapper.py -file query2\_reducer.py

**Mapper:**

**A screenshot of a computer code

Description automatically generated**

**Reducer: Output:**

**A screenshot of a computer code

Description automatically generated**A screenshot of a white screen

Description automatically generated

Query 3)

**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/part.tbl -output /data/output3 -mapper query3\_mapper.py -reducer query3\_reducer.py -file query3\_mapper.py -file query3\_reducer.py

**Mapper:**

**A screenshot of a computer code

Description automatically generated**

**Reducer:**

**A screen shot of a computer code

Description automatically generated**

**Output:**

A number and numbers on a white background

Description automatically generated

Query 4)

**Command line:**

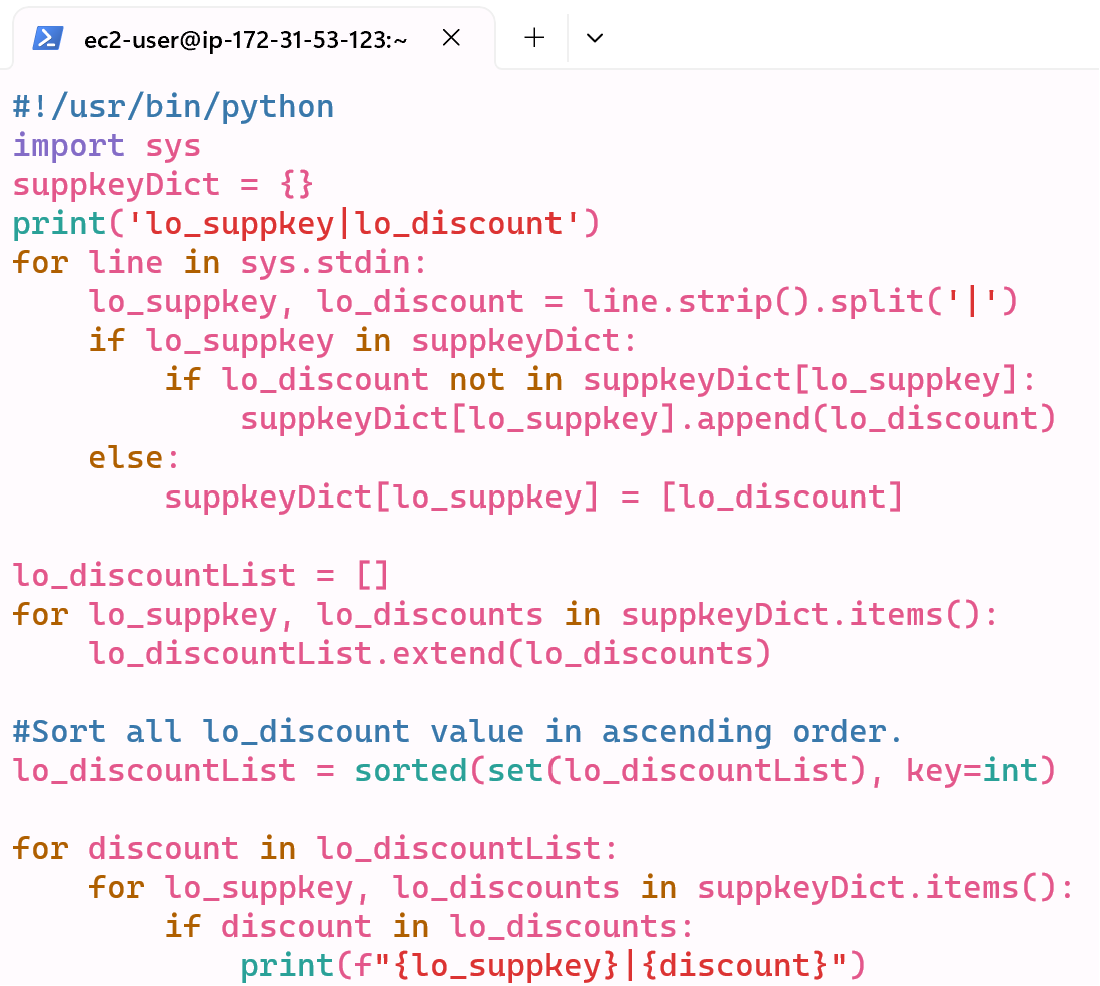
hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl -output /data/output4 -mapper query4\_mapper.py -reducer query4\_reducer.py -file query4\_mapper.py -file query4\_reducer.py

**Mapper:**

A screenshot of a computer code

Description automatically generated

**Reducer: Output:**

**A screenshot of a computer code

Description automatically generated**

Query 5)

**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl -output /data/output5 -mapper query5\_mapper.py -reducer query5\_reducer.py -file query5\_mapper.py -file query5\_reducer.py

**Mapper:**

**A screenshot of a computer code

Description automatically generated**

**Reducer:**

**A screenshot of a computer code

Description automatically generated**

**Output:**

A screenshot of a computer

Description automatically generated

Query 6)

**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl -output /data/output6 -mapper query6\_mapper.py -reducer query6\_reducer.py -file query6\_mapper.py -file query6\_reducer.py

**Mapper:**

A screenshot of a computer code

Description automatically generated

**Reducer: Output:**

A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated

Query 7) Total 150 outputs



**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl,/data/dwdate.tbl -output /data/output7 -mapper query7\_mapper.py -reducer query7\_reducer.py -file query7\_mapper.py -file query7\_reducer.py

**Mapper:**

**A screenshot of a computer code

Description automatically generated**

**Reducer: Output:**

**A screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated**

Query 8) Total 214 rows.



**Command line:**

hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl,/data/dwdate.tbl -output /data/output8 -mapper query8\_mapper.py -reducer query8\_reducer.py -file query8\_mapper.py -file query8\_reducer.py

**Mapper:**

**A screenshot of a computer code

Description automatically generated**

**Reducer:**

**A screenshot of a computer program

Description automatically generated**

**Output:**

**A screenshot of a computer

Description automatically generated**