Student Name: Mai Ngo

Course Name and Number: CSC 555 Big Data Mining

Assignment 3

Date: 10/25/2023

**Part 1)** Multi-node cluster

**Question 1**

Verify that the cluster is running: <http://34.227.172.169:50070/>

A screenshot of a computer

Description automatically generated

**Question 2 -** Wordcount example

Screenshot of Wordcount Map-Reduce running: 3-node.

A white paper with pink text

Description automatically generated

Screenshot of Wordcount Map-Reduce running: 1-node.

A close-up of a text

Description automatically generated

Time:

|  |  |
| --- | --- |
| 1 Node | 3-Nodes |
|  | **A number of numbers on a white background  Description automatically generated** |

Discussion:

The bioproject.xml file has a size of 231149003 bytes or 220.29 MB – from Hadoop Streaming job: ‘Number of splits: 2’. The Real execution time on one node is approximately 50% longer compared to using 3-nodes, 54.251 sec and 37.831 sec respectively. This is because 3-nodes cluster allows for parallel processing, the two blocks will be processed simultaneously. While 1 node cluster will have each block processed sequentially. Moreover, both User and Sys times are the same, i.e., CPU time; this indicates both clusters are computationally efficient. Overall, the time difference shows the advantages of distributed computing, i.e., parallelism. Using 3-nodes can cut execution time effectively, improve processing performance when working with large data files.

**Question 3 –** Assignment 2 – Part 2

|  |  |  |
| --- | --- | --- |
|  | 1-Node | 3-Nodes |
| Query 1 |  |  |
| Discussion – 1st Query: Running time are the same for first Query due to input file size. Supplier table has a file size of 166676 bytes = 0.159 MB, which will be distributed to only one block. Given this, only one node will process the data. Thus, running time will be somewhat the same. | | |
| Query 2 |  |  |
| Discussion – 2nd Query: Running time on a 1-node is approximately double compared to using 3-nodes, Lineorder table has a file size of 594313001 bytes = 556.74 MB, which is distributed across five blocks (size 128MB). Same explanation as part 2, paralellism allows faster processing performance. Using 3-nodes cluster, each execution round can have 3 blocks processsed simutaneously. | | |
| Query 3 |  |  |
| Discussion – 3rd Query: Running time are the same due to input file size. Part table has a file size of 17139259 bytes = 16.33 MB. Same explanation as 1st Query. | | |
| Query 4 |  |  |
| Query 5 |  |  |
| Query 6 |  |  |
| Discussion – 4th, 5th, 6th Queries: Same explanation as 2nd Query, same input table Lineorder. | | |
| Query 7 |  |  |
| Query 8 |  |  |
| Discussion – 7th, 8th Queries: Same explanation as 2nd Query, given 2 input tables Lineorder and Dwdate; which has a file size of 229965 bytes = 0.22 MB. The total input size is 556.96 MB. which is still distributed across five blocks (size 128MB). | | |

**Part 2)** Two-Step MapReduce

Implement, execute, and time the following query using Hadoop streaming with python.  
SELECT lo\_quantity, MAX(lo\_revenue)  
FROM (SELECT lo\_revenue, MAX(lo\_quantity) as lo\_quantity,  
MAX(lo\_discount) as lo\_discount  
FROM lineorder  
WHERE lo\_orderpriority LIKE '%URGENT'  
GROUP BY lo\_revenue)  
WHERE lo\_discount BETWEEN 5 AND 8  
GROUP BY lo\_quantity;

**Note:** I comment further on each code in .ipynb/.py/.html file. Breaking down big query for 2 MapReduce.

**Mapper 1:**

**A screenshot of a computer code

Description automatically generated**

**Reducer 1:**

**A screenshot of a computer code

Description automatically generated**

**Mapper 2:**

**A screenshot of a computer

Description automatically generated**

**Reducer 2:**

**A screenshot of a computer code

Description automatically generated**

**Command line and time:**

**1st MR:** time hadoop jar hadoop-streaming-2.6.4.jar -input /data/lineorder.tbl -output /data/MR1\_Output -mapper HW3\_mapper1.py -reducer HW3\_reducer1.py -file HW3\_mapper1.py -file HW3\_reducer1.py

**A white background with red text

Description automatically generated**

**2nd MR:** time hadoop jar hadoop-streaming-2.6.4.jar -input /data/MR1\_Output/part-00000 -output /data/MR2\_Output -mapper HW3\_mapper2.py -reducer HW3\_reducer2.py -file HW3\_mapper2.py -file HW3\_reducer2.py

**A white background with red text

Description automatically generated**

**Total Time:** 47.937 sec

**Output:** Total 50 outputs.

****

**A screenshot of a computer

Description automatically generated**