Terna Engineering College
Computer Engineering Department

Program: Sem VII

Course: Big Data Analytics & Computational Lab -I (BDA&CL-I)

Experiment No. 05

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: BE-COMPS-50	Batch: B3
Date of Experiment: 05-10-2021	Date of Submission: 05-10-2021
Grade :	

Aim: To implement matrix multiplication using MapReduce.

B.1 Software code written by a student:

➔ **Mapper.py**

```
#!/usr/bin/env python
import sys
m_r=2
m_c=3
n_r=3
n_c=2
i=0
for line in sys.stdin:
        el=map(int,line.split())
        if(i<m_r):
                for j in range(len(el)):
                        for k in range(n_c):
                                print('%d\t%d\t%d\t%d' %(i,k,j,el[j]))
        else:
                for j in range(len(el)):
                        for k in range(m_r):
                                print('%d\t%d\t%d\t%d' %(k,j,i+m_r,el[j]))
        i=i+1
```

➔ **Reducer.py**

```
#!/usr/bin/env python
import sys
m_r=2
m_c=3
n_r=3
n_c=2

matrix=[]
for row in range(m_r):
        r=[]
        for col in range(n_c):
                s=0
                for el in range(m_c):
                        mul=1
                        for num in range(2):
                                line=sys.stdin.readline()
                                n = map(int,line.split('\t'))[-1]
                                mul*=n
                        s+=mul
                r.append(s)
        matrix.append(r)
print('\n'.join([str(x) for x in matrix]))
```

B.2 Input and Output:

Input:

➔ Matrix1.txt

1	2	3
4	5	6

➔ Matrix2.txt

7	8
5	6
10	11

Output:

```
[biadmin@bivm ~]$ cd matrix
[biadmin@bivm matrix]$ nano matrix1.txt
[biadmin@bivm matrix]$ nano matrix2.txt
[biadmin@bivm matrix]$ nano mapper.py
[biadmin@bivm matrix]$ cat *.txt |python mapper.py
0       0       0       1
0       1       0       1
0       0       1       2
0       1       1       2
0       0       2       3
0       1       2       3
1       0       0       4
1       1       0       4
1       0       1       5
1       1       1       5
1       0       2       6
1       1       2       6
0       0       4       7
1       0       4       7
0       1       4       8
1       1       4       8
0       0       5       5
1       0       5       5
0       1       5       6
1       1       5       6
0       0       6       10
1       0       6       10
0       1       6       11
1       1       6       11
[biadmin@bivm matrix]$ cat *.txt |python mapper.py|python reducer.py
[14, 77]
[138, 257]
[biadmin@bivm matrix]$
```

B.3 Observations and learning:

- ➔ A MapReduce job usually splits the input datasets and then processes each of them independently by the Map tasks in a completely parallel manner.
- ➔ The output is then sorted and input to reduce tasks. Both job input and output are stored in file systems.
- ➔ Tasks are scheduled and monitored by the framework.

B.4 Conclusion:

Hence we've successfully implemented a program to implement matrix multiplication using Map Reduce in Hadoop Ecosystem.

B.5 Question of Curiosity:

1. What is shuffling and sorting in map-reduce?

Ans:

Shuffling in MapReduce:

- ➔ The process of transferring data from the mappers to reducers is shuffling. It is also the process by which the system performs the sort. Then it transfers the map output to the reducer as input. This is the reason the shuffle phase is necessary for the reducers.
- ➔ Otherwise, they would not have any input (or input from every mapper). Since shuffling can start even before the map phase has finished. So this saves some time and completes the tasks in lesser time

Sorting in MapReduce:

- ➔ MapReduce Framework automatically sorts the keys generated by the mapper. Thus, before starting of reducer, all intermediate key-value pairs get sorted by key and not by value. It does not sort values passed to each reducer. They can be in any order.
- ➔ This saves time for the reducer. Reducer in MapReduce starts a new reduce task when the next key in the sorted input data is different than the previous. Each reduced task takes key-value pairs as input and generates key-value pairs as output.

2. What are the functions of mapper and reducer?

Ans:

- ➔ Mappers and Reducers are java processes that are created on the worker node of HDFS tor process data. Mappers typically read data and Reducers aggregates the data.
- ➔ MapReduce is a programming model to process a high volume of data in a distributed fashion.
- ➔ In the context of HDFS, a framework is developed using the MapReduce programming model to process data stored in HDFS. It is known as Hadoop MapReduce. This framework is developed using Java and it provides API for developing business logic for Mappers and Reducers as needed.
- ➔ The framework also takes care of distributing the mappers and reducers on the cluster and monitoring it till the execution of a job.