Department of Computer Engineering
Academic Term: July-November 2023

Rubrics for Lab Experiments

Class : B.*E. Computer* **Subject Name :*BDA***
Semester : VII **Subject Code :**

Practical No:	4
Title:	Write a program to implement Matrix Multiplication algorithm using Map Reduce
Date of Performance:	03-09-2023
Roll No:	9427
Name of the Student:	Atharva Prashant Pawar

Evaluation:

Performance Indicator	Below average	Average	Good	Excellent	Marks
On time Submission (2)	Not submitted(0)	Submitted after deadline (1)	Early or on time submission(2)	---	
Test cases and output (4)	Incorrect output (1)	The expected output is verified only a for few test cases (2)	The expected output is Verified for all test cases but is not presentable (3)	Expected output is obtained for all test cases. Presentable and easy to follow (4)	
Coding efficiency (2)	The code is not structured at all (0)	The code is structured but not efficient (1)	The code is structured and efficient. (2)	-	
Knowledge(2)	Basic concepts not clear (0)	Understood the basic concepts (1)	Could explain the concept with suitable example (1.5)	Could relate the theory with real world application(2)	
Total					

Signature of the Teacher :

Code:

MatrixMultiplication.java

```java
package matrix;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class MatrixMultiplication {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("m", "2");
        conf.set("n", "5");
        conf.set("p", "3");

        Job job = new Job(conf, "MatrixMultiplication");

        job.setJarByClass(MatrixMultiplication.class);

        job.setOutputKeyClass(Text.class);

        job.setOutputValueClass(Text.class);

        job.setMapperClass(MatrixMapper.class);

        job.setReducerClass(MatrixReducer.class);

        job.setInputFormatClass(TextInputFormat.class);

        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));

        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.waitForCompletion(true);
    }
}
```

Java source file | length : 1,195 lines : 40 | Ln : 40 Col : 2 Pos : 1,196 | Windows (CR LF) | UTF-8 | INS

MatrixMultiplication.java | MatrixMapper.java

```java
package matrix;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MatrixMapper extends Mapper<LongWritable, Text, Text, Text> {
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

        Configuration conf = context.getConfiguration();

        int m = Integer.parseInt(conf.get("m"));
        int p = Integer.parseInt(conf.get("p"));

        String line = value.toString();
        String[] indicesAndValue = line.split(",");

        Text outputKey = new Text();
        Text outputValue = new Text();

        if (indicesAndValue[0].equals("A")) {
            for (int k = 0; k < p; k++) {
                outputKey.set(indicesAndValue[1] + "," + k);
                outputValue.set("A," + indicesAndValue[2] + "," +
                indicesAndValue[3]);
                context.write(outputKey, outputValue);
            }
        } else {
            for (int i = 0; i < m; i++) {
                outputKey.set(i + "," + indicesAndValue[2]);
                outputValue.set("B," + indicesAndValue[1] + "," +
                indicesAndValue[3]);
                context.write(outputKey, outputValue);
            }
        }
    }
}
```

MatrixMultiplication.java | MatrixMapper.java | MatrixReducer.java

```java
package matrix;
import java.io.IOException;
import java.util.HashMap;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class MatrixReducer extends Reducer<Text, Text, Text, Text> {
    public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
        String[] value;
        HashMap<Integer, Float> hashA = new HashMap<Integer,Float>();
        HashMap<Integer, Float> hashB = new HashMap<Integer,Float>();

        for (Text val : values) {
            value = val.toString().split(",");
            if (value[0].equals("A")) {
                hashA.put(Integer.parseInt(value[1]),Float.parseFloat(value[2]));
            } else {
                hashB.put(Integer.parseInt(value[1]),Float.parseFloat(value[2]));
            }
        }
        int n = Integer.parseInt(context.getConfiguration().get("n"));
        float result = 0.0f;
        float a_ij;
        float b_jk;

        for (int j = 0; j < n; j++) {
            a_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f;
            b_jk = hashB.containsKey(j) ? hashB.get(j) : 0.0f;
            result += a_ij * b_jk;
        }

        if (result != 0.0f) {
            context.write(null, new Text(key.toString() + "," + Float.toString(result)));
        }
    }
}
```

OUTPUT:

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /matrix_multiplication
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -mkdir /matrix_multiplication/Input

[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -put /home/cloudera/Desktop/windows_files/input.txt /matrix_multiplication/In
put
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -ls /matrix_multiplication/Input
Found 1 items
-rw-r--r--   1 hdfs supergroup        223 2023-08-07 10:03 /matrix_multiplication/Input/input.txt
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -cat /matrix_multiplication/Input/input.txt
A,0,0,4
A,0,1,5
A,0,2,6
A,0,3,7
A,0,4,8
A,1,0,1
A,1,1,3
A,1,2,4
A,1,3,5
A,1,4,6
B,0,0,6
B,0,1,3
B,0,2,4
B,1,0,1
B,1,1,2
B,1,2,7
B,2,0,8
B,2,1,6
B,2,2,5
B,3,0,2
B,3,1,3
B,3,2,1
B,4,0,7
B,4,1,4
B,4,2,2[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop jar /home/cloudera/Desktop/windows_files/MatrixMultiplicationNew.jar /matrix_mul
tiplication/Input /matrix_multiplication/Output
23/08/07 10:04:48 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
23/08/07 10:04:48 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
23/08/07 10:04:48 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface
 and execute your application with ToolRunner to remedy this.
23/08/07 10:04:49 INFO input.FileInputFormat: Total input paths to process : 1
23/08/07 10:04:49 INFO mapreduce.JobSubmitter: number of splits:1
23/08/07 10:04:49 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2124456538_0001
23/08/07 10:04:49 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
23/08/07 10:04:49 INFO mapreduce.Job: Running job: job_local2124456538_0001
23/08/07 10:04:49 INFO mapred.LocalJobRunner: OutputCommitter set in config null
23/08/07 10:04:49 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
23/08/07 10:04:49 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
23/08/07 10:04:49 INFO mapred.LocalJobRunner: Waiting for map tasks
23/08/07 10:04:49 INFO mapred.LocalJobRunner: Starting task: attempt_local2124456538_0001_m_000000_0
23/08/07 10:04:49 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
23/08/07 10:04:49 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
23/08/07 10:04:49 INFO mapred.MapTask: Processing split: hdfs://quickstart.cloudera:8020/matrix_multiplication/Input/input.tx
t:0+223
23/08/07 10:04:49 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
23/08/07 10:04:49 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
23/08/07 10:04:49 INFO mapred.MapTask: soft limit at 83886080
23/08/07 10:04:49 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
23/08/07 10:04:49 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
23/08/07 10:04:49 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
23/08/07 10:04:49 INFO mapred.LocalJobRunner:
23/08/07 10:04:49 INFO mapred.MapTask: Starting flush of map output
23/08/07 10:04:49 INFO mapred.MapTask: Spilling map output
```

```
23/08/07 10:04:49 INFO mapred.Task: Task:attempt_local2124456538_0001_m_000000_0 is done. And is in the process of committing
23/08/07 10:04:49 INFO mapred.LocalJobRunner: map
23/08/07 10:04:49 INFO mapred.Task: Task 'attempt_local2124456538_0001_m_000000_0' done.
23/08/07 10:04:49 INFO mapred.LocalJobRunner: Finishing task: attempt_local2124456538_0001_m_000000_0
23/08/07 10:04:49 INFO mapred.LocalJobRunner: map task executor complete.
23/08/07 10:04:49 INFO mapred.LocalJobRunner: Waiting for reduce tasks
23/08/07 10:04:49 INFO mapred.LocalJobRunner: Starting task: attempt_local2124456538_0001_r_000000_0
23/08/07 10:04:49 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
23/08/07 10:04:49 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
23/08/07 10:04:49 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@ea5867
1
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=180564784, maxSingleShuffleLimit=45141196, mergeTh
reshold=119172760, ioSortFactor=10, memToMemMergeOutputsThreshold=10
23/08/07 10:04:49 INFO reduce.EventFetcher: attempt_local2124456538_0001_r_000000_0 Thread started: EventFetcher for fetching
 Map Completion Events
23/08/07 10:04:49 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt_local2124456538_0001_m_0000
00_0 decomp: 722 len: 726 to MEMORY
23/08/07 10:04:49 INFO reduce.InMemoryMapOutput: Read 722 bytes from map-output for attempt_local2124456538_0001_m_000000_0
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 722, inMemoryMapOutputs.size() -> 1,
 commitMemory -> 0, usedMemory ->722
23/08/07 10:04:49 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
23/08/07 10:04:49 INFO mapred.LocalJobRunner: 1 / 1 copied.
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
23/08/07 10:04:49 INFO mapred.Merger: Merging 1 sorted segments
23/08/07 10:04:49 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 716 bytes
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: Merged 1 segments, 722 bytes to disk to satisfy reduce memory limit
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: Merging 1 files, 726 bytes from disk
23/08/07 10:04:49 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce
23/08/07 10:04:49 INFO mapred.Merger: Merging 1 sorted segments
23/08/07 10:04:49 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 716 bytes
23/08/07 10:04:49 INFO mapred.LocalJobRunner: 1 / 1 copied.
23/08/07 10:04:49 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords
23/08/07 10:04:49 INFO mapred.Task: Task:attempt_local2124456538_0001_r_000000_0 is done. And is in the process of committing
23/08/07 10:04:49 INFO mapred.LocalJobRunner: 1 / 1 copied.
23/08/07 10:04:49 INFO mapred.Task: Task attempt_local2124456538_0001_r_000000_0 is allowed to commit now
23/08/07 10:04:49 INFO output.FileOutputCommitter: Saved output of task 'attempt_local2124456538_0001_r_000000_0' to hdfs://q
uickstart.cloudera:8020/matrix_multiplication/Output/_temporary/0/task_local2124456538_0001_r_000000
23/08/07 10:04:49 INFO mapred.LocalJobRunner: reduce > reduce
23/08/07 10:04:49 INFO mapred.Task: Task 'attempt_local2124456538_0001_r_000000_0' done.
```

```
23/08/07 10:04:49 INFO mapred.LocalJobRunner: reduce task executor complete.
23/08/07 10:04:50 INFO mapreduce.Job: Job job_local2124456538_0001 running in uber mode : false
23/08/07 10:04:50 INFO mapreduce.Job:  map 100% reduce 100%
23/08/07 10:04:50 INFO mapreduce.Job: Job job_local2124456538_0001 completed successfully
23/08/07 10:04:50 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=11878
                FILE: Number of bytes written=543268
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=446
                HDFS: Number of bytes written=57
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=25
                Map output records=60
                Map output bytes=600
                Map output materialized bytes=726
                Input split bytes=134
                Combine input records=0
                Combine output records=0
                Reduce input groups=6
                Reduce shuffle bytes=726
                Reduce input records=60
                Reduce output records=6
                Spilled Records=120
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=515899392
        Shuffle Errors
                BAD_ID=0
```

```
B,4,2,2[cloudera@quickstasudo -u hdfs hadoop fs -ls /matrix_multiplication/Output
Found 2 items
-rw-r--r--   1 hdfs supergroup          0 2023-08-07 10:04 /matrix_multiplication/Output/_SUCCESS
-rw-r--r--   1 hdfs supergroup         57 2023-08-07 10:04 /matrix_multiplication/Output/part-r-00000
```

```
[cloudera@quickstart ~]$ sudo -u hdfs hadoop fs -cat /matrix_multiplication/Output/part-r-00000
0,0,147.0
0,1,111.0
0,2,104.0
1,0,93.0
1,1,72.0
1,2,62.0
```

Post lab

Atharva Prashant Pawar (9427) -[Batch-D]

BDA Exp 4 DATE:

Q1. Define what is block in HDFS?

=>

Block in HDFS:

HDFS (Hadoop Distributed File System) is the primary storage system used in Hadoop for storing large datasets accross a clusters of machines.

HDFS divides files into smaller units known as 'blocks' to facilitate efficient storage & processing.

Blocks are the fundamental storage unit in HDFS, & their size is typically set to 128 MB or 256 MB, although it can be configured.

Unlike traditional file system that might store a file as a whole, HDFS breaks files into fixed-size blocks for several reasons.

Q2. Why is a block in HDFS so large?

=> A. Efficient Disk I/O:

Larger blocks result in fewer metadata operations (like opening / closing files) per unit of data, reducing overhead.

Reading or writing a large block at once improves disk I/O efficiency compared to smaller chunks.

B

DATE:

B. Reduced N/W overhead:
Data transfer involves n/w communication b/w DataNodes & clients. Larger blocks amortize the n/w overhead over more data, improving efficiency.
Smaller blocks might result in a higher percentage of overhead due to frequent data transfer initiation

C. Optimal for Sequential Access:
HDFS is designed for applications with large files & mostly sequential access patterns.
Larger blocks suit this design by allowing for continuous reading / writing of data, reducing seek time & improving throughput.

D. Minimized Metadata Overhead:
HDFS NameNode maintains metadata about files, & larger blocks mean fewer file records to manage.
Smaller blocks would lead to more metadata entries for the same amount of data, increasing the NameNode's load.

E. Better Load Distribution:
Large blocks help distribute data uniformly across DataNodes, preventing hotspots where certain nodes become overloaded due to small block sizes.