Big Data Analytics Innovative Assignment Report

NIRMA
UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Presented to: Prof. Aparna Kumari

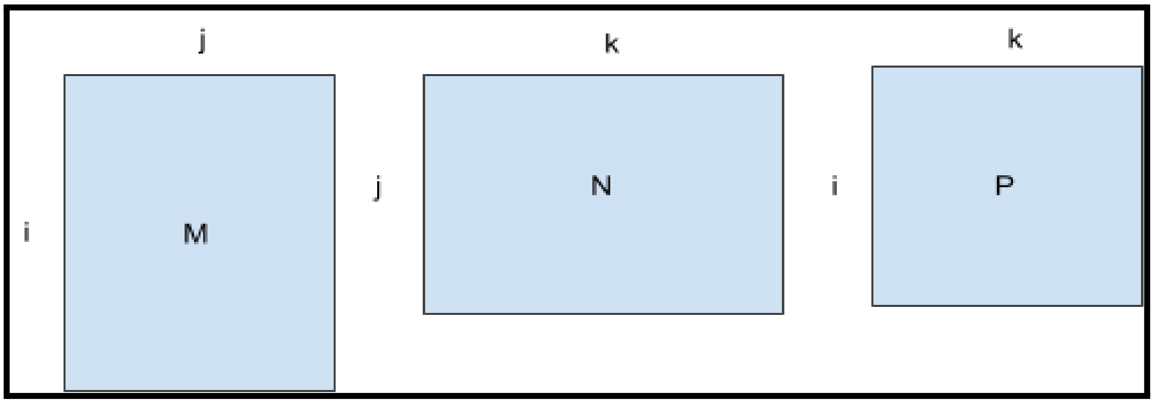
Presented by: Dip Patel (19BCE166)
Jalpan Patel(19BCE177)
Parv Patel(19BCE190)

Sparse Matrix Multiplication with Hadoop

INTRODUCTION

Matrix multiplication is a fundamental operation in linear algebra with related real-life applications, such as matrix factorization, chemical system formulation, and graph analysis [14]. In addition to its naturally related applications, several problems are reducible by matrix multiplication. Thus, these problems should be investigated thoroughly to enhance the efficiency of implemented algorithms for matrix multiplication. Given the inputs of two matrices A and B, where the number of columns in A equals the number of rows in B, matrix multiplication produces matrix C with the number of rows equal to that in A and number of columns equal to that in B. The Brute-Force matrix multiplication algorithm for square matrices is given in Algorithm 1. The Brute-Force algorithm has a high processing complexity of O(n3), but suffers from the massive memory lookup process required to locate each array element for multiplication. Over the years, several matrix multiplication algorithms have been proposed to reduce the cost and time
of the matrix multiplication process [2, 15].

Matrix Multiplication is when we have a **“p x q”** matrix **M**, whose element in **row i** and **column j** will be denoted **m_{ij}** and a **“q x r”** matrix **N** whose element in **row j** and **column k** is denoted by **n_{jk}** then the product **P = MN** will be **“p x r”** matrix **P** whose element in **row i** and **column k** will be donated by **P_{ik}**, where **P(i,k)** = **m_{ij}*n_{jk}**

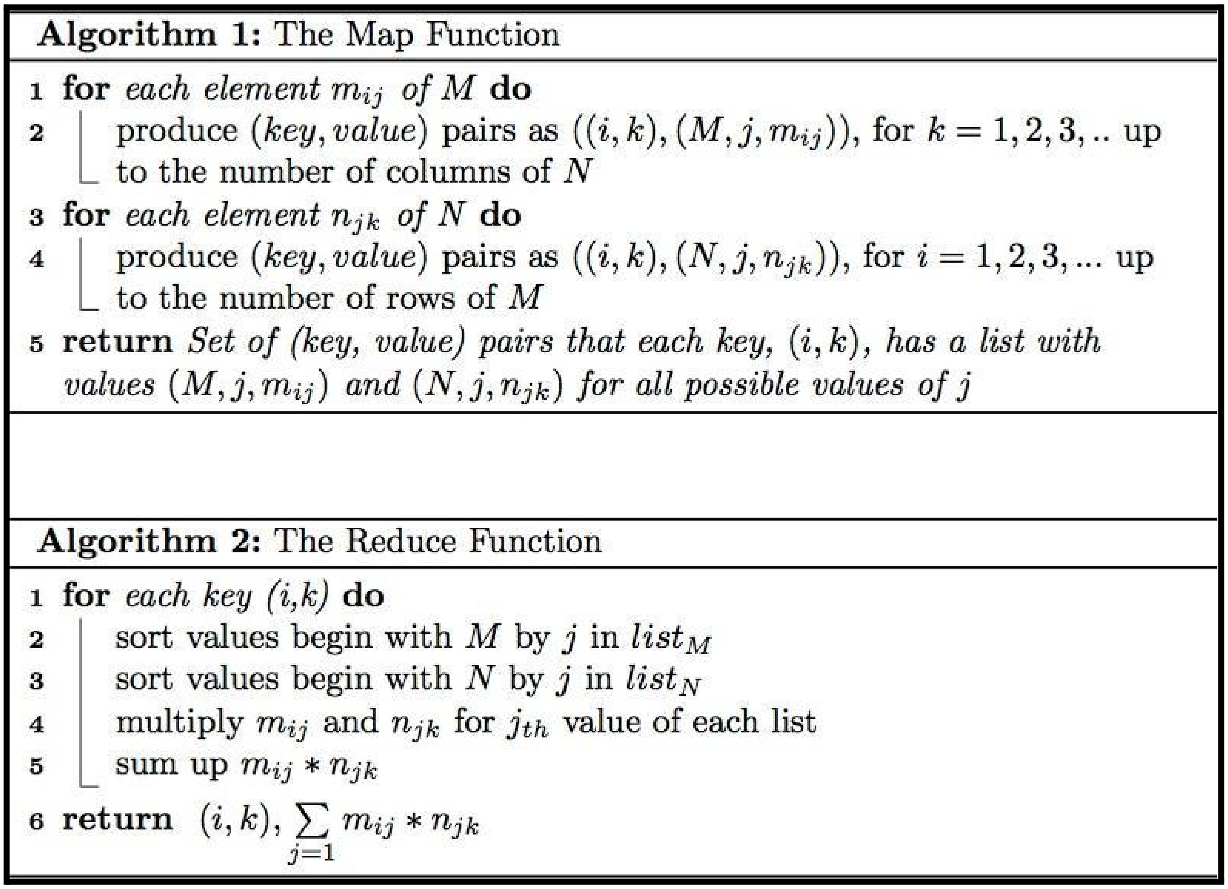


Most matrices are sparse so large numbers of cells have value zero. When we represent matrices in this form, we do not need to keep entries for the cells that have values of zero to

save a large amount of disk space. As input data files, we store matrix M and N on HDFS in the following format:

MAP-REDUCE

The map and reduce functions have been built using the following Algorithms:



```
Algorithm 1: The Map Function
for each element $m_{ij}$ of $M$ do
    produce $(key, value)$ pairs as $((i,k),(M,j,m_{ij}))$, for $k = 1,2,3,..$ up to the number of columns of $N$
for each element $n_{jk}$ of $N$ do
    produce $(key, value)$ pairs as $((i,k),(N,j,n_{jk}))$, for $i = 1,2,3,...$ up to the number of rows of $M$
return Set of (key, value) pairs that each key, $(i,k)$, has a list with values $(M,j,m_{ij})$ and $(N,j,n_{jk})$ for all possible values of $j$
```

```
Algorithm 2: The Reduce Function
for each key (i,k) do
    sort values begin with $M$ by $j$ in $list_M$
    sort values begin with $N$ by $j$ in $list_N$
    multiply $m_{ij}$ and $n_{jk}$ for $j_{th}$ value of each list
    sum up $m_{ij} * n_{jk}$
return $(i,k), \sum_{j=1} m_{ij} * n_{jk}$
```

Map function will produce key,value pairs from the input data as it is described in Algorithm 1. Reduce function uses the output of the Map function and performs the calculations and produces key,value pairs as described in Algorithm 2. All outputs are written to HDFS.

1. **MAP Task:**
    a. **For matrix M**: Map task (Algorithm 1) will produce key,value pairs as

        follows: (i,k), (M, j, m_{ij})

        m_{11} = 1
        (1,1), (M, 1, 1) for k = 1
        (1,2), (M, 1, 1) for k = 2

        m_{12} = 1
        (1,1), (M, 2, 1) for k = 1
        (1,2), (M, 2, 1) for k = 2

m_{13} = 1
(1,1), (M, 3, 1) for k = 1
(1,2), (M, 3, 1) for k = 2

m_{21} = 1
(2,1), (M, 1, 1) for k = 1
(2,2), (M, 1, 1) for k = 2

m_{22} = 1
(2,1), (M, 2, 1) for k = 1
(2,2), (M, 2, 1) for k = 2

m_{23} = 1
(2,1), (M, 3, 1) for k = 1
(2,2), (M, 3, 1) for k = 2

b. **For Matrix N:** Map task (Algorithm 1) will produce key, value pairs as

follows: (i,k), (N, j, n_{jk})

n_{11} = 1
(1,1), (N, 1, 1) for i = 1
(2,1), (N, 1, 1) for i = 2

n_{21} = 1
(1,1), (N, 2, 1) for i = 1
(2,1), (N, 2, 1) for i = 2

n_{31} = 1
(1,1), (N, 3, 1) for i = 1
(2,1), (N, 3, 1) for i = 2

n_{12} = 2
(1,2), (N, 1, 1) for i = 1
(2,2), (N, 1, 1) for i = 2

n_{22} = 1
(1,2), (N, 2, 1) for i = 1
(2,2), (N, 2, 1) for i = 2

n_{32} = 1
(1,2), (N, 3, 1) for i = 1
(2,2), (N, 3, 1) for i = 2

c. **After combine operation the map task will return key, value pairs will look like as follows:**

((i,k), [(M, j, mij), (M, j, mij),..., (N, j, njk), (N, j, njk),   ])

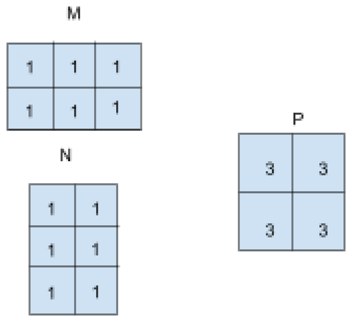
(
(1,1), [(M, 1, 1), (M, 2, 1), (M, 3, 1), (N, 1, 1), (N, 2, 1), (N, 3, 1)]
(1,2), [(M, 1, 1), (M, 2, 1), (M, 3, 1), (N, 1, 1), (N, 2, 1), (N, 3, 1)]
(2,1), [(M, 1, 1), (M, 2, 1), (M, 3, 1), (N, 1, 1), (N, 2, 1), (N, 3, 1)]
(2,1), [(M, 1, 1), (M, 2, 1), (M, 3, 1), (N, 1, 1), (N, 2, 1), (N, 3, 1)]
)

Note that the entries for the same key are grouped in the same list, which is performed by the framework. This output will be stored in HDFS and is fed to the reducer task as input.

2. **REDUCER Task:**

Reduce task takes the key,value pairs as the input and processes one key at a time. For each key it divides the values in two separate lists for M and N. As an example, it will create the following list for key (1,1), [(M, 1, 1), (M, 2, 1), (M, 3, 1), (N, 1, 1), (N, 2, 1), (N, 3, 1)]
I_m = [(M, 1, 1), (M, 2, 1), (M, 3, 1)]
I_n = [(N, 1, 1), (N, 2, 1), (N, 3, 1)]
then sums up the multiplication of m_{ij} and n_{jk} for each j as follows:
P(1,1) = 1 + 1 + 1 = 3
P(1,2) = 1 + 1 + 1 = 3
P(2,1) = 1 + 1 + 1 = 3
P(2,2) = 1 + 1 + 1 = 3

Upon completing all calculations we will get the following output:



M

1	1	1
1	1	1

N

1	1
1	1
1	1

P

3	3
3	3

Hadoop

Hadoop is an Apache open-source platform for storing, processing, and analyzing massive amounts of data. Hadoop is a Java-based data warehouse that is not an OLAP system (online analytical processing). It's a batch/offline processing system. It could also be upscaled by adding extra nodes to the cluster.

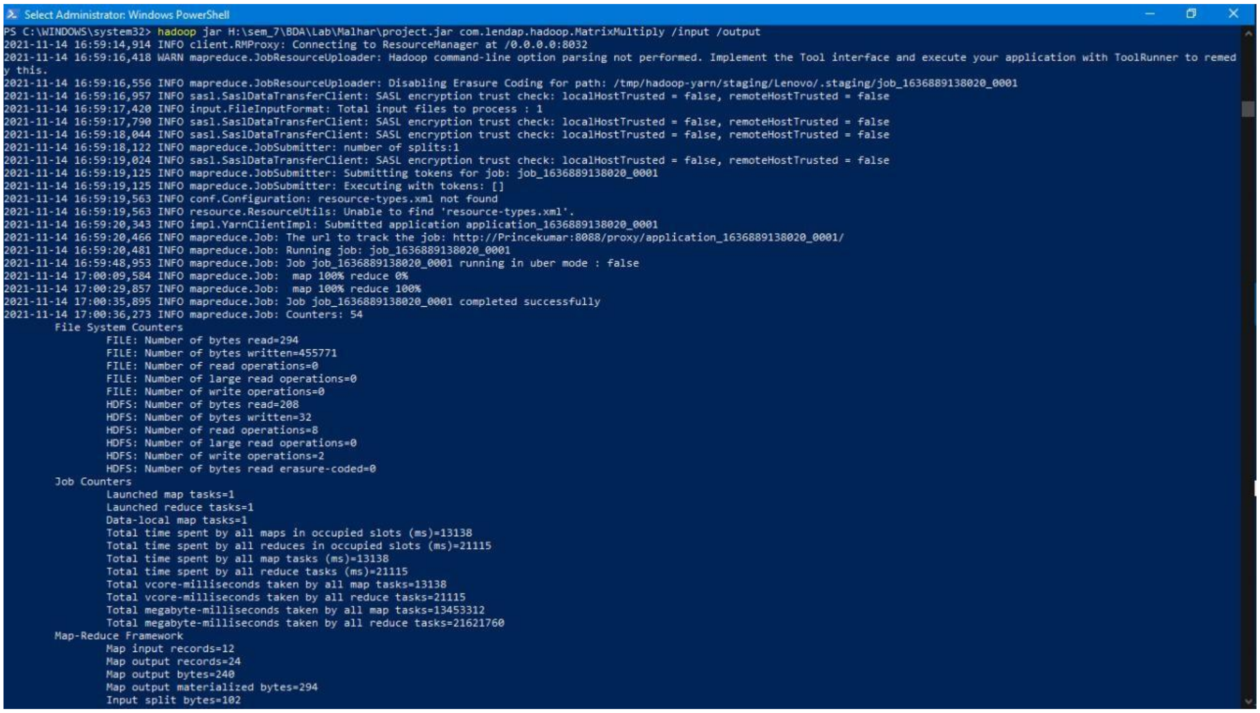
Modules of Hadoop

- HDFS: Hadoop Distributed File System. Google published a document called GFS, and HDFS was developed on the basis of it. The data will be divided down into blocks and stored in nodes as part of the distributed design, according to the document.
- Yarn: Yet another Resource Negotiator is utilized to control the cluster and schedule jobs.
- Map Reduce: This is a framework that allows Java programs to perform parallel computing on data employing key-value pairs. The Map task turns input data into a data set that can be evaluated in Key-value pairs. The result of the Map activity is absorbed by the Reduce task, which subsequently outputs the required result.
- Hadoop Common: These Java libraries are utilized by several other Hadoop modules and can be used to launch Hadoop

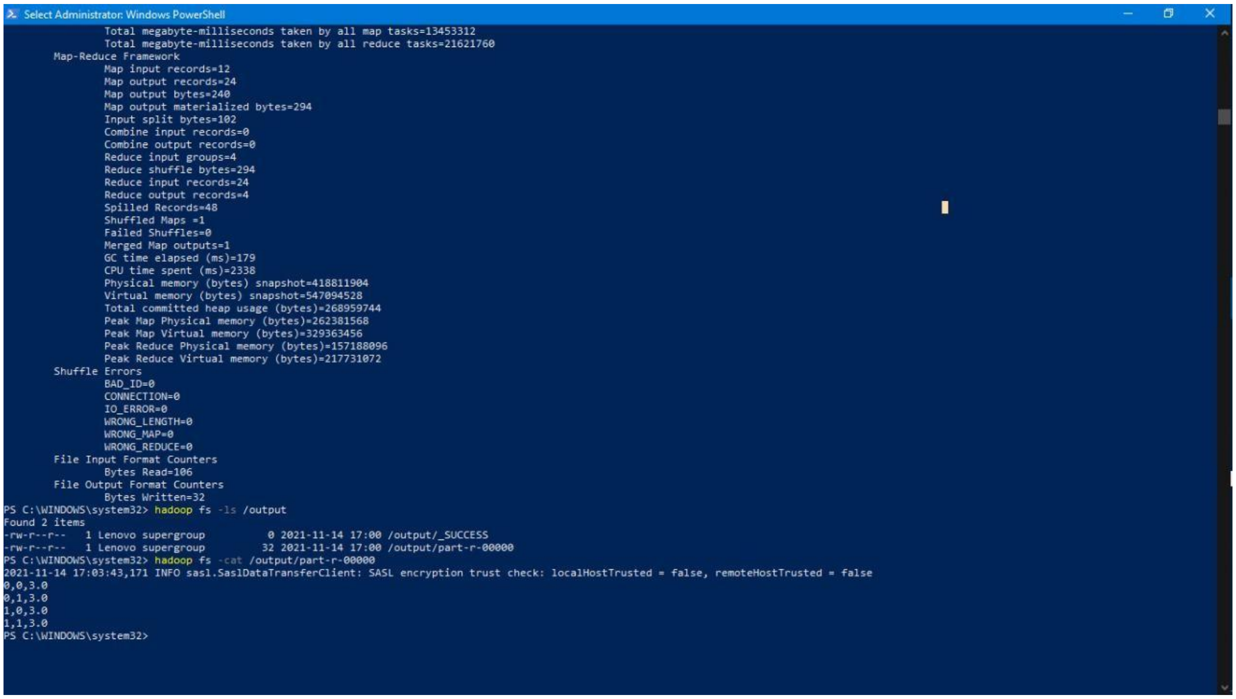
Specifications

Title	**Description**
Data specification	Matrices of various sizes are tested with dimensions in a range up to 10^4.
Configuration of System	**RAM -** 8GB **Processor -** Intel Core i5 8th Gen
Cluster setup	**Single node -** One DataNode running and setting up all the NameNode, DataNode, ResourceManager, and NodeManager on a single machine.
Parameters	**Time comparison -** So in our comparison when we ran the MP with 1 reducer the task of all reducers was completed in 69052 ms, but when we increased the number of reducers to 5 reducers the task of all reducers was completed in 152296 ms. **Sparsity -** Sparsity of 1000*1000 input was .3 which means that around 30% of it was filled with non-zero values, the rest was filled with 0s.

Input-Output

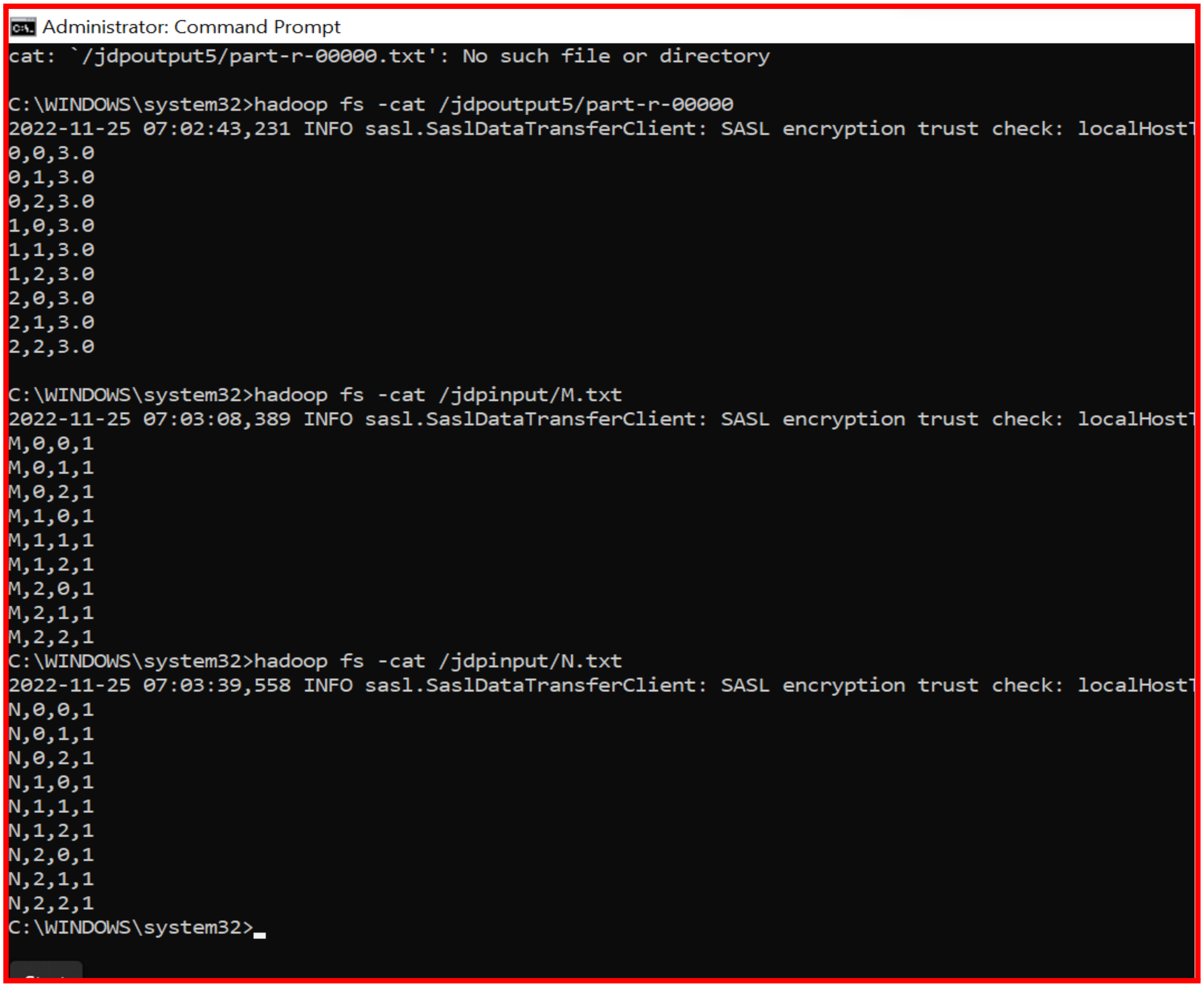


```
PS C:\WINDOWS\system32> hadoop jar H:\sem_7\BDA\Lab\Malhar\project.jar com.lendap.hadoop.MatrixMultiply /input /output
2021-11-14 16:59:14,914 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-14 16:59:16,418 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remed
y this.
2021-11-14 16:59:16,556 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Lenovo/.staging/job_1636889138020_0001
2021-11-14 16:59:16,957 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-14 16:59:17,420 INFO input.FileInputFormat: Total input files to process : 1
2021-11-14 16:59:17,790 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-14 16:59:18,044 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-14 16:59:18,122 INFO mapreduce.JobSubmitter: number of splits:1
2021-11-14 16:59:19,024 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-14 16:59:19,125 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1636889138020_0001
2021-11-14 16:59:19,125 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-14 16:59:19,563 INFO conf.Configuration: resource-types.xml not found
2021-11-14 16:59:19,563 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-11-14 16:59:20,343 INFO impl.YarnClientImpl: Submitted application application_1636889138020_0001
2021-11-14 16:59:20,466 INFO mapreduce.Job: The url to track the job: http://Princekumar:8088/proxy/application_1636889138020_0001/
2021-11-14 16:59:20,481 INFO mapreduce.Job: Running job: job_1636889138020_0001
2021-11-14 16:59:48,953 INFO mapreduce.Job: Job job_1636889138020_0001 running in uber mode : false
2021-11-14 17:00:09,584 INFO mapreduce.Job:  map 100% reduce 0%
2021-11-14 17:00:29,857 INFO mapreduce.Job:  map 100% reduce 100%
2021-11-14 17:00:35,895 INFO mapreduce.Job: Job job_1636889138020_0001 completed successfully
2021-11-14 17:00:36,273 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=294
                FILE: Number of bytes written=455771
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=208
                HDFS: Number of bytes written=32
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=13138
                Total time spent by all reduces in occupied slots (ms)=21115
                Total time spent by all map tasks (ms)=13138
                Total time spent by all reduce tasks (ms)=21115
                Total vcore-milliseconds taken by all map tasks=13138
                Total vcore-milliseconds taken by all reduce tasks=21115
                Total megabyte-milliseconds taken by all map tasks=13453312
                Total megabyte-milliseconds taken by all reduce tasks=21621760
        Map-Reduce Framework
                Map input records=12
                Map output records=24
                Map output bytes=240
                Map output materialized bytes=294
                Input split bytes=102
```



```
                Total megabyte-milliseconds taken by all map tasks=13453312
                Total megabyte-milliseconds taken by all reduce tasks=21621760
        Map-Reduce Framework
                Map input records=12
                Map output records=24
                Map output bytes=240
                Map output materialized bytes=294
                Input split bytes=102
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=294
                Reduce input records=24
                Reduce output records=4
                Spilled Records=48
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=179
                CPU time spent (ms)=2338
                Physical memory (bytes) snapshot=418811904
                Virtual memory (bytes) snapshot=547094528
                Total committed heap usage (bytes)=268959744
                Peak Map Physical memory (bytes)=262381568
                Peak Map Virtual memory (bytes)=329363456
                Peak Reduce Physical memory (bytes)=157188096
                Peak Reduce Virtual memory (bytes)=217731072
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=106
        File Output Format Counters
                Bytes Written=32
PS C:\WINDOWS\system32> hadoop fs -ls /output
Found 2 items
-rw-r--r--   1 Lenovo supergroup          0 2021-11-14 17:00 /output/_SUCCESS
-rw-r--r--   1 Lenovo supergroup         32 2021-11-14 17:00 /output/part-r-00000
PS C:\WINDOWS\system32> hadoop fs -cat /output/part-r-00000
2021-11-14 17:03:43,171 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
0,0,3.0
0,1,3.0
1,0,3.0
1,1,3.0
PS C:\WINDOWS\system32>
```

Changed output



```
cat: `/jdpoutput5/part-r-00000.txt': No such file or directory

C:\WINDOWS\system32>hadoop fs -cat /jdpoutput5/part-r-00000
2022-11-25 07:02:43,231 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostT
0,0,3.0
0,1,3.0
0,2,3.0
1,0,3.0
1,1,3.0
1,2,3.0
2,0,3.0
2,1,3.0
2,2,3.0

C:\WINDOWS\system32>hadoop fs -cat /jdpinput/M.txt
2022-11-25 07:03:08,389 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostT
M,0,0,1
M,0,1,1
M,0,2,1
M,1,0,1
M,1,1,1
M,1,2,1
M,2,0,1
M,2,1,1
M,2,2,1
C:\WINDOWS\system32>hadoop fs -cat /jdpinput/N.txt
2022-11-25 07:03:39,558 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostT
N,0,0,1
N,0,1,1
N,0,2,1
N,1,0,1
N,1,1,1
N,1,2,1
N,2,0,1
N,2,1,1
N,2,2,1
C:\WINDOWS\system32>
```

Applications:

1) Perspective projections, which are the core of 3D animation, are created using matrix multiplication. A two-dimensional image depicting a position in 3-dimensional space is displayed on the computer display. You may switch across two and three-dimensional worlds using matrix multiplication.
2) It's frequently utilized in fields like network theory, linear system of equations, coordinate system transformation, and population modeling, to list some of them.
3) Matrix multiplication was first established in linear algebra to make calculations easier and more clear. Matrix multiplication and linear algebra have a close link that is vital in all of the math, along with chemistry, physics, engineering, and computer science.
4) In graph theory, matrix multiplication — especially, powers of a given matrix A — is a valuable tool when the matrix in hand is the adjacency matrix of a graph.
5) Matrix multiplication is a fundamental concept in quantum mechanics and physics in general. The moment of inertia tensor, Hamiltonians-based continuous-time representations of the development of physical systems, and the most general formulation of the Lorentz transformation from special relativity are some examples.
6) In mathematical finance, matrices are widely employed in a number of ways. A

correlation matrix, for example, is a table in which an item (i,j) indicates the degree to which price movements in item I and item j are associated during a given timeframe. Every day, a large number of computer cycles are expended calculating and analyzing such types of matrices in order to, in part, estimate the risk involved with such a basket of items.

7) Any linear system of equations, i.e. a plane in certain N-dimensional space, can be described by a matrix. All current physics and simulations require multiplying one plane with another plane.
    a) If you must determine the gravitational force across a path, the path matrix is multiplied by the gravity matrix.
    b) If the intensity of a magnetic field (Matrix a) at a spatial position described by matrix B must be determined, the magnetic field is multiplied by the geographical matrix.
    c) If you want to locate the best location in town for a firehouse, an As matrix A, you create a block-by-block fire hazard matrix, and as matrix B, you create a trip time matrix centered on a given place in town.
    d) To create a self-driving automobile employing neural networks in the computer vision system, rely on GPUs to interpret images employing real-time matrix multiplication.

Challenges

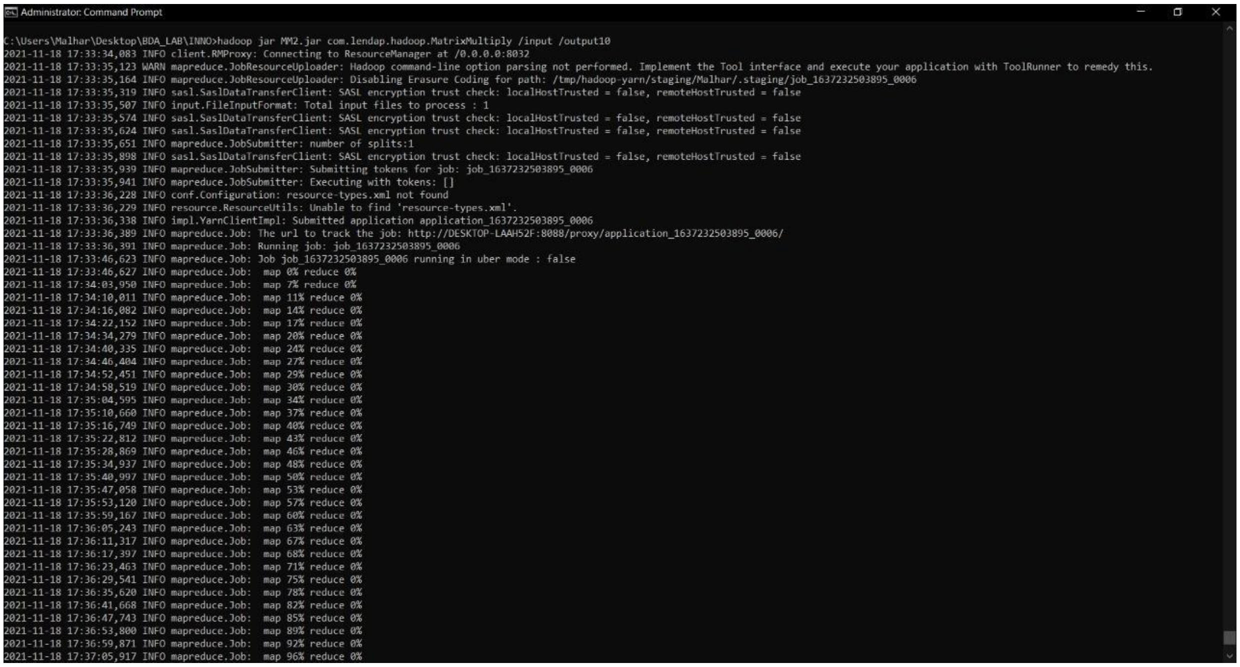
Three issues unique to the sparse scenario are identified.

- The first is a problem with processor load imbalance. In the dense state, when W = N3, every processor produces a net of W/p work, but in the sparse state, such a proportion is difficult to attain. Additionally, dense GEMM algorithms are often executed in s stages, with every processor performing W/p s work per stage. It's significantly more difficult to achieve appropriate load balance per stage for sparse matrices than it is to achieve load balance for the entire operation. Considering matrices with complete diagonal, asynchronous techniques may be useful in alleviating the load balancing issue.
- The addition of submatrices poses the second problem. For submatrix additions, no additional operations are done in the dense case; the same amount of work would be required in the sequential case.
- The third difficulty is to keep the communication hidden. While this is simpler in the dense case because of the higher computation to communication proportion, it is more difficult in the sparse scenario. To reduce the communication to computation ratio of sparse algorithms, simply increasing the problem size is usually insufficient.

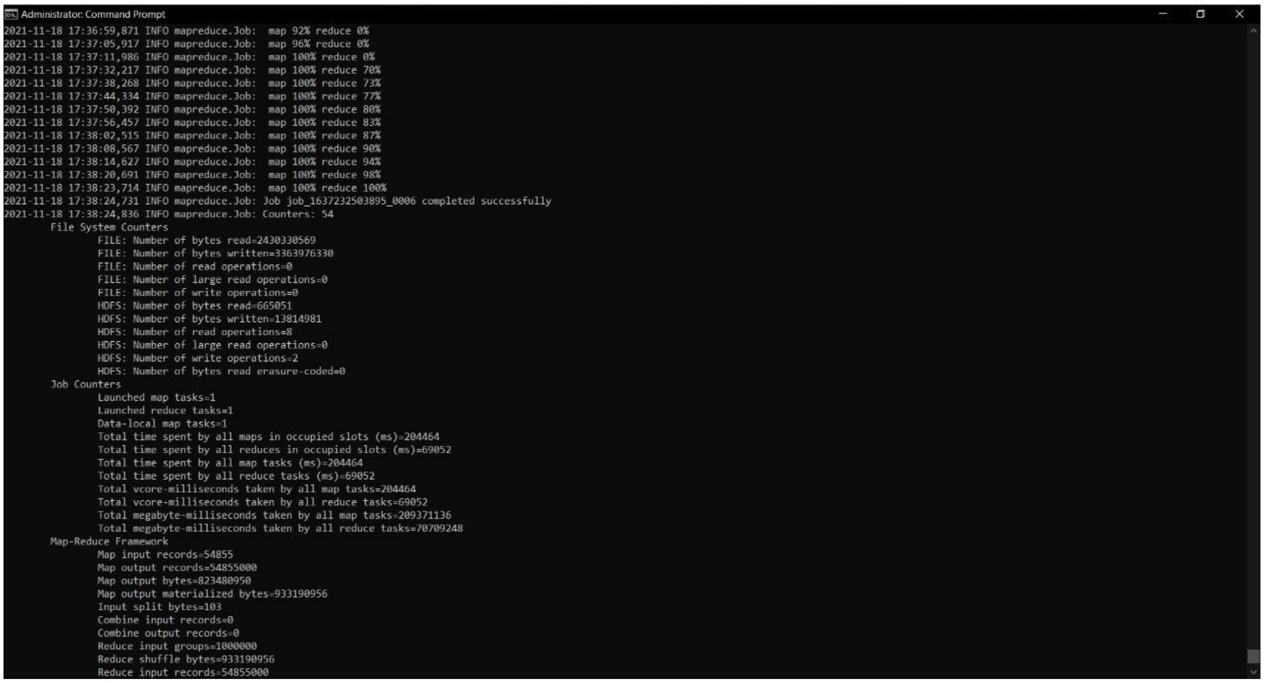
Result/Observations

- Matrix of large dimensions

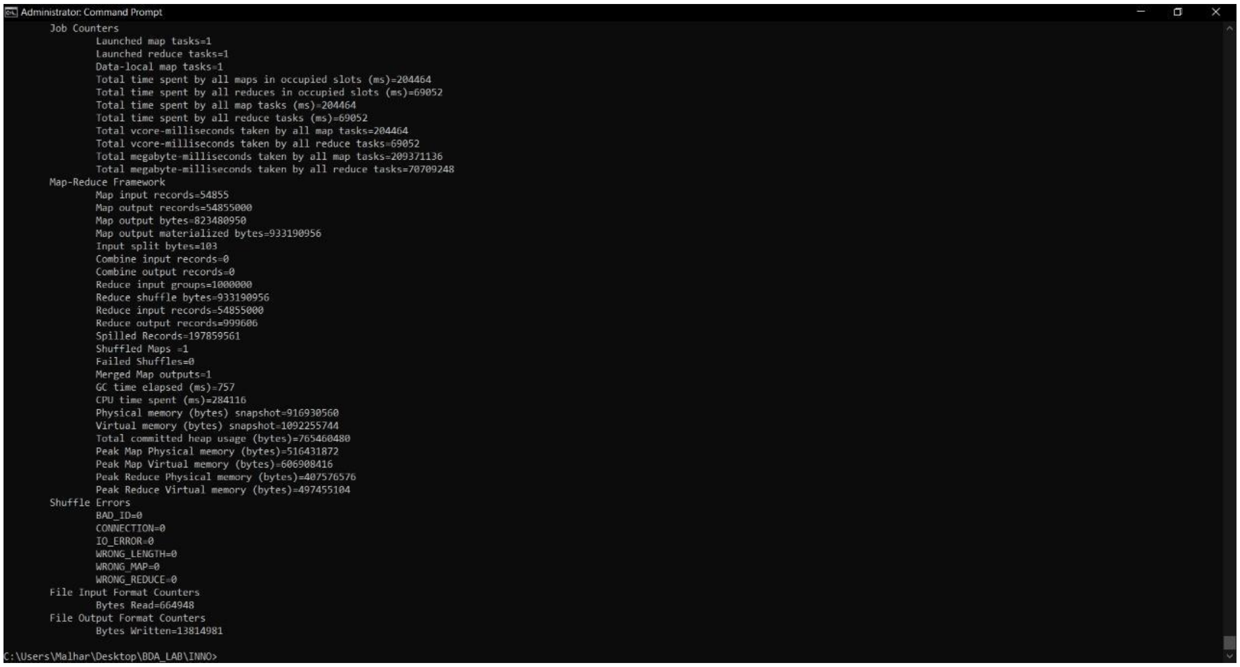
On taking 1000*100 size of M, 100*1000 size of N we received 1000*1000 size of P, the time taken for execution was a couple of minutes.



```
C:\Users\Malhar\Desktop\BDA_LAB\INNO>hadoop jar MM2.jar com.lendap.hadoop.MatrixMultiply /input /output10
2021-11-18 17:33:34,083 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-18 17:33:35,123 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-11-18 17:33:35,164 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Malhar/.staging/job_1637232503895_0006
2021-11-18 17:33:35,319 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 17:33:35,507 INFO input.FileInputFormat: Total input files to process : 1
2021-11-18 17:33:35,574 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 17:33:35,624 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 17:33:35,651 INFO mapreduce.JobSubmitter: number of splits:1
2021-11-18 17:33:35,898 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 17:33:35,939 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1637232503895_0006
2021-11-18 17:33:35,941 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-18 17:33:36,228 INFO conf.Configuration: resource-types.xml not found
2021-11-18 17:33:36,229 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-11-18 17:33:36,338 INFO impl.YarnClientImpl: Submitted application application_1637232503895_0006
2021-11-18 17:33:36,389 INFO mapreduce.Job: The url to track the job: http://DESKTOP-LAAH52F:8088/proxy/application_1637232503895_0006/
2021-11-18 17:33:36,391 INFO mapreduce.Job: Running job: job_1637232503895_0006
2021-11-18 17:33:46,623 INFO mapreduce.Job: Job job_1637232503895_0006 running in uber mode : false
2021-11-18 17:33:46,627 INFO mapreduce.Job:  map 0% reduce 0%
2021-11-18 17:34:03,950 INFO mapreduce.Job:  map 7% reduce 0%
2021-11-18 17:34:10,011 INFO mapreduce.Job:  map 11% reduce 0%
2021-11-18 17:34:16,082 INFO mapreduce.Job:  map 14% reduce 0%
2021-11-18 17:34:22,152 INFO mapreduce.Job:  map 17% reduce 0%
2021-11-18 17:34:34,279 INFO mapreduce.Job:  map 20% reduce 0%
2021-11-18 17:34:40,335 INFO mapreduce.Job:  map 24% reduce 0%
2021-11-18 17:34:46,404 INFO mapreduce.Job:  map 27% reduce 0%
2021-11-18 17:34:52,451 INFO mapreduce.Job:  map 29% reduce 0%
2021-11-18 17:34:58,519 INFO mapreduce.Job:  map 30% reduce 0%
2021-11-18 17:35:04,595 INFO mapreduce.Job:  map 34% reduce 0%
2021-11-18 17:35:10,660 INFO mapreduce.Job:  map 37% reduce 0%
2021-11-18 17:35:16,749 INFO mapreduce.Job:  map 40% reduce 0%
2021-11-18 17:35:22,812 INFO mapreduce.Job:  map 43% reduce 0%
2021-11-18 17:35:28,869 INFO mapreduce.Job:  map 46% reduce 0%
2021-11-18 17:35:34,937 INFO mapreduce.Job:  map 48% reduce 0%
2021-11-18 17:35:40,997 INFO mapreduce.Job:  map 50% reduce 0%
2021-11-18 17:35:47,058 INFO mapreduce.Job:  map 53% reduce 0%
2021-11-18 17:35:53,120 INFO mapreduce.Job:  map 57% reduce 0%
2021-11-18 17:35:59,167 INFO mapreduce.Job:  map 60% reduce 0%
2021-11-18 17:36:05,243 INFO mapreduce.Job:  map 63% reduce 0%
2021-11-18 17:36:11,317 INFO mapreduce.Job:  map 67% reduce 0%
2021-11-18 17:36:17,397 INFO mapreduce.Job:  map 68% reduce 0%
2021-11-18 17:36:23,463 INFO mapreduce.Job:  map 71% reduce 0%
2021-11-18 17:36:29,541 INFO mapreduce.Job:  map 75% reduce 0%
2021-11-18 17:36:35,620 INFO mapreduce.Job:  map 78% reduce 0%
2021-11-18 17:36:41,668 INFO mapreduce.Job:  map 82% reduce 0%
2021-11-18 17:36:47,743 INFO mapreduce.Job:  map 85% reduce 0%
2021-11-18 17:36:53,800 INFO mapreduce.Job:  map 89% reduce 0%
2021-11-18 17:36:59,871 INFO mapreduce.Job:  map 92% reduce 0%
2021-11-18 17:37:05,917 INFO mapreduce.Job:  map 96% reduce 0%
```

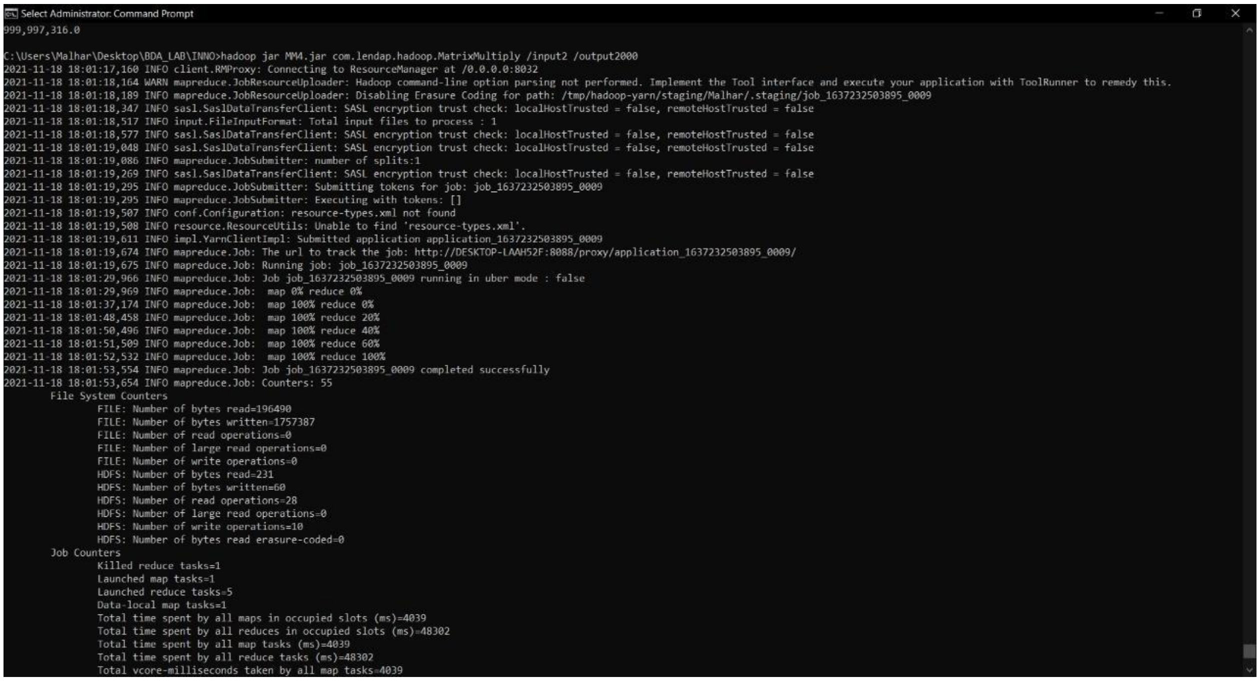


```
2021-11-18 17:36:59,871 INFO mapreduce.Job:  map 92% reduce 0%
2021-11-18 17:37:05,917 INFO mapreduce.Job:  map 96% reduce 0%
2021-11-18 17:37:11,986 INFO mapreduce.Job:  map 100% reduce 0%
2021-11-18 17:37:32,217 INFO mapreduce.Job:  map 100% reduce 70%
2021-11-18 17:37:38,268 INFO mapreduce.Job:  map 100% reduce 73%
2021-11-18 17:37:44,334 INFO mapreduce.Job:  map 100% reduce 77%
2021-11-18 17:37:50,392 INFO mapreduce.Job:  map 100% reduce 80%
2021-11-18 17:37:56,457 INFO mapreduce.Job:  map 100% reduce 83%
2021-11-18 17:38:02,515 INFO mapreduce.Job:  map 100% reduce 87%
2021-11-18 17:38:08,567 INFO mapreduce.Job:  map 100% reduce 90%
2021-11-18 17:38:14,627 INFO mapreduce.Job:  map 100% reduce 94%
2021-11-18 17:38:20,691 INFO mapreduce.Job:  map 100% reduce 98%
2021-11-18 17:38:23,714 INFO mapreduce.Job:  map 100% reduce 100%
2021-11-18 17:38:24,731 INFO mapreduce.Job: Job job_1637232503895_0006 completed successfully
2021-11-18 17:38:24,836 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=2430330569
                FILE: Number of bytes written=3363976330
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=665051
                HDFS: Number of bytes written=13814981
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=204464
                Total time spent by all reduces in occupied slots (ms)=69052
                Total time spent by all map tasks (ms)=204464
                Total time spent by all reduce tasks (ms)=69052
                Total vcore-milliseconds taken by all map tasks=204464
                Total vcore-milliseconds taken by all reduce tasks=69052
                Total megabyte-milliseconds taken by all map tasks=209371136
                Total megabyte-milliseconds taken by all reduce tasks=70709248
        Map-Reduce Framework
                Map input records=54855
                Map output records=54855000
                Map output bytes=823480950
                Map output materialized bytes=933190956
                Input split bytes=103
                Combine input records=0
                Combine output records=0
                Reduce input groups=1000000
                Reduce shuffle bytes=933190956
                Reduce input records=54855000
```

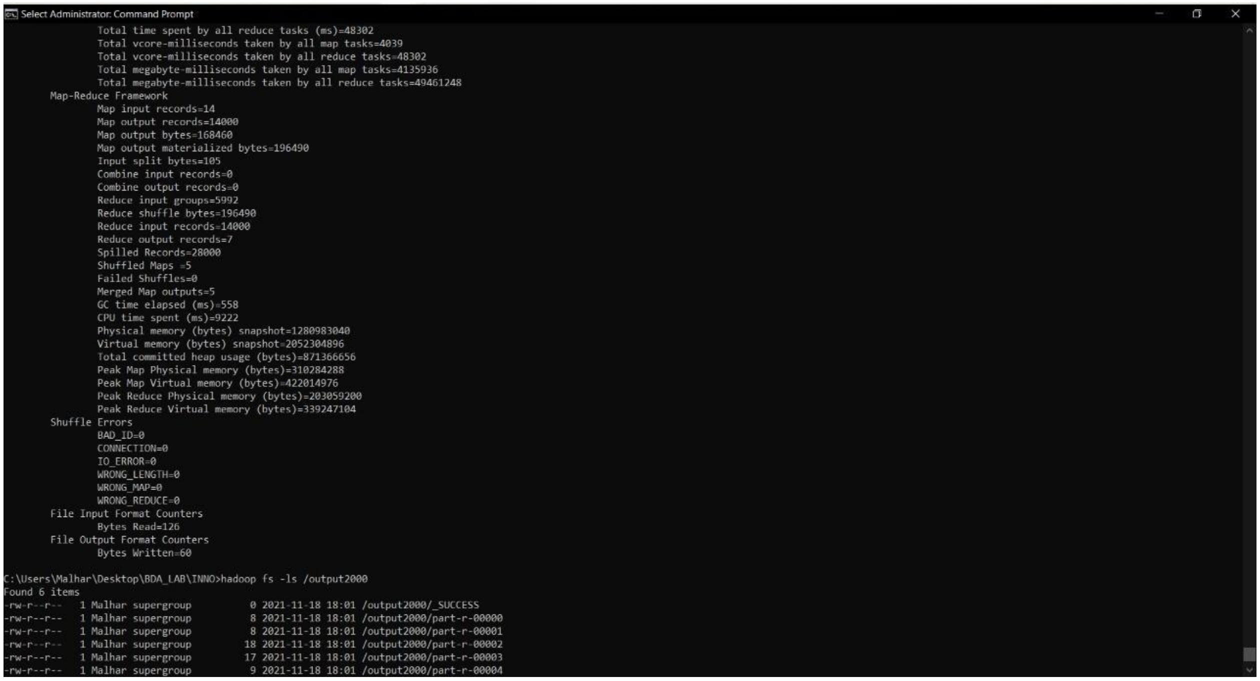


```
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=204464
                Total time spent by all reduces in occupied slots (ms)=69052
                Total time spent by all map tasks (ms)=204464
                Total time spent by all reduce tasks (ms)=69052
                Total vcore-milliseconds taken by all map tasks=204464
                Total vcore-milliseconds taken by all reduce tasks=69052
                Total megabyte-milliseconds taken by all map tasks=209371136
                Total megabyte-milliseconds taken by all reduce tasks=70709248
        Map-Reduce Framework
                Map input records=54855
                Map output records=54855000
                Map output bytes=823480950
                Map output materialized bytes=933190956
                Input split bytes=103
                Combine input records=0
                Combine output records=0
                Reduce input groups=1000000
                Reduce shuffle bytes=933190956
                Reduce input records=54855000
                Reduce output records=999606
                Spilled Records=197859561
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=757
                CPU time spent (ms)=284116
                Physical memory (bytes) snapshot=916930560
                Virtual memory (bytes) snapshot=1092255744
                Total committed heap usage (bytes)=765460480
                Peak Map Physical memory (bytes)=516431872
                Peak Map Virtual memory (bytes)=606908416
                Peak Reduce Physical memory (bytes)=407576576
                Peak Reduce Virtual memory (bytes)=497455104
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=664948
        File Output Format Counters
                Bytes Written=13814981

C:\Users\Malhar\Desktop\BDA_LAB\INNO>
```

- The number of Reducer increased by 5



```
999,997,316.0

C:\Users\Malhar\Desktop\BDA_LAB\INNO>hadoop jar MM4.jar com.lendap.hadoop.MatrixMultiply /input2 /output2000
2021-11-18 18:01:17,160 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2021-11-18 18:01:18,164 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-11-18 18:01:18,189 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Malhar/.staging/job_1637232503895_0009
2021-11-18 18:01:18,347 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 18:01:18,517 INFO input.FileInputFormat: Total input files to process : 1
2021-11-18 18:01:18,577 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 18:01:19,048 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 18:01:19,086 INFO mapreduce.JobSubmitter: number of splits:1
2021-11-18 18:01:19,269 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2021-11-18 18:01:19,295 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1637232503895_0009
2021-11-18 18:01:19,295 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-11-18 18:01:19,507 INFO conf.Configuration: resource-types.xml not found
2021-11-18 18:01:19,508 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-11-18 18:01:19,611 INFO impl.YarnClientImpl: Submitted application application_1637232503895_0009
2021-11-18 18:01:19,674 INFO mapreduce.Job: The url to track the job: http://DESKTOP-LAAH52F:8088/proxy/application_1637232503895_0009/
2021-11-18 18:01:19,675 INFO mapreduce.Job: Running job: job_1637232503895_0009
2021-11-18 18:01:29,966 INFO mapreduce.Job: Job job_1637232503895_0009 running in uber mode : false
2021-11-18 18:01:29,969 INFO mapreduce.Job:  map 0% reduce 0%
2021-11-18 18:01:37,174 INFO mapreduce.Job:  map 100% reduce 0%
2021-11-18 18:01:48,458 INFO mapreduce.Job:  map 100% reduce 20%
2021-11-18 18:01:50,496 INFO mapreduce.Job:  map 100% reduce 40%
2021-11-18 18:01:51,509 INFO mapreduce.Job:  map 100% reduce 60%
2021-11-18 18:01:52,532 INFO mapreduce.Job:  map 100% reduce 100%
2021-11-18 18:01:53,554 INFO mapreduce.Job: Job job_1637232503895_0009 completed successfully
2021-11-18 18:01:53,654 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=196490
                FILE: Number of bytes written=1757387
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=231
                HDFS: Number of bytes written=60
                HDFS: Number of read operations=28
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=10
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed reduce tasks=1
                Launched map tasks=1
                Launched reduce tasks=5
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=4039
                Total time spent by all reduces in occupied slots (ms)=48302
                Total time spent by all map tasks (ms)=4039
                Total time spent by all reduce tasks (ms)=48302
                Total vcore-milliseconds taken by all map tasks=4039
```



```
                Total time spent by all reduce tasks (ms)=48302
                Total vcore-milliseconds taken by all map tasks=4039
                Total vcore-milliseconds taken by all reduce tasks=48302
                Total megabyte-milliseconds taken by all map tasks=4135936
                Total megabyte-milliseconds taken by all reduce tasks=49461248
        Map-Reduce Framework
                Map input records=14
                Map output records=14000
                Map output bytes=168460
                Map output materialized bytes=196490
                Input split bytes=105
                Combine input records=0
                Combine output records=0
                Reduce input groups=5992
                Reduce shuffle bytes=196490
                Reduce input records=14000
                Reduce output records=7
                Spilled Records=28000
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=558
                CPU time spent (ms)=9222
                Physical memory (bytes) snapshot=1280983040
                Virtual memory (bytes) snapshot=2052304896
                Total committed heap usage (bytes)=871366656
                Peak Map Physical memory (bytes)=310284288
                Peak Map Virtual memory (bytes)=422014976
                Peak Reduce Physical memory (bytes)=203059200
                Peak Reduce Virtual memory (bytes)=339247104
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=126
        File Output Format Counters
                Bytes Written=60

C:\Users\Malhar\Desktop\BDA_LAB\INNO>hadoop fs -ls /output2000
Found 6 items
-rw-r--r--   1 Malhar supergroup          0 2021-11-18 18:01 /output2000/_SUCCESS
-rw-r--r--   1 Malhar supergroup          8 2021-11-18 18:01 /output2000/part-r-00000
-rw-r--r--   1 Malhar supergroup          8 2021-11-18 18:01 /output2000/part-r-00001
-rw-r--r--   1 Malhar supergroup         18 2021-11-18 18:01 /output2000/part-r-00002
-rw-r--r--   1 Malhar supergroup         17 2021-11-18 18:01 /output2000/part-r-00003
-rw-r--r--   1 Malhar supergroup          9 2021-11-18 18:01 /output2000/part-r-00004
```

So, it is usually preferred to keep the number of reducers such that it is:

- It is a multiple of the block size
- It does not take a long time to execute

- ○ Doesn’t require a lot of file creations

➢ But we also know that the number of reducers can’t exceed the number of partitions, so we also need to take that into consideration when deciding the number of reducers. In the instance, we make the number of reducers 10 but we have a single partition system then the output will be divided into 10 shards but those 10 outputs will be consolidated by a single reducer.

➢ The benefits of having more reducers are as follows:
- ○ Increases load balancing
- ○ Lowers the cost of failure

➢ But on the other hand, having too many reducers can also:
- ○ Increase the framework overhead
- ○ Slower start-up time
- ○ Increases the number of input for the next tasks

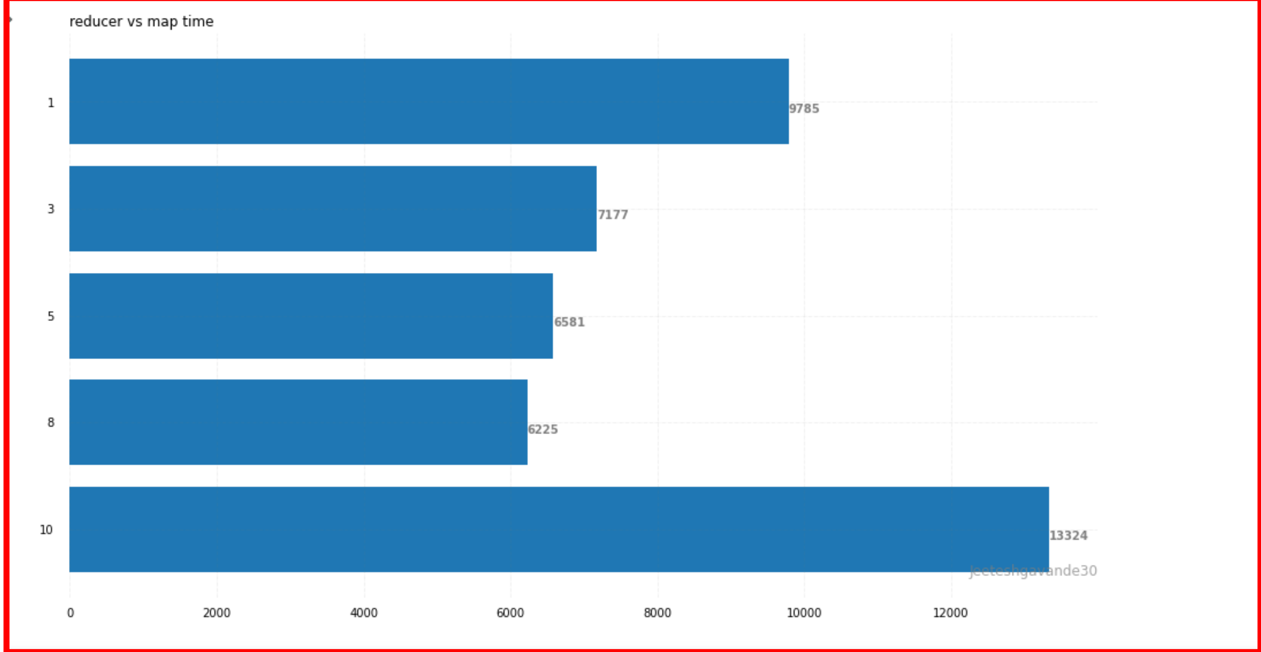
Conclusion

The Hadoop framework's MapReduce component is a critical processing component. It's a rapid, scalable, and cost-effective tool that may assist data analysts and developers in processing large amounts of information.

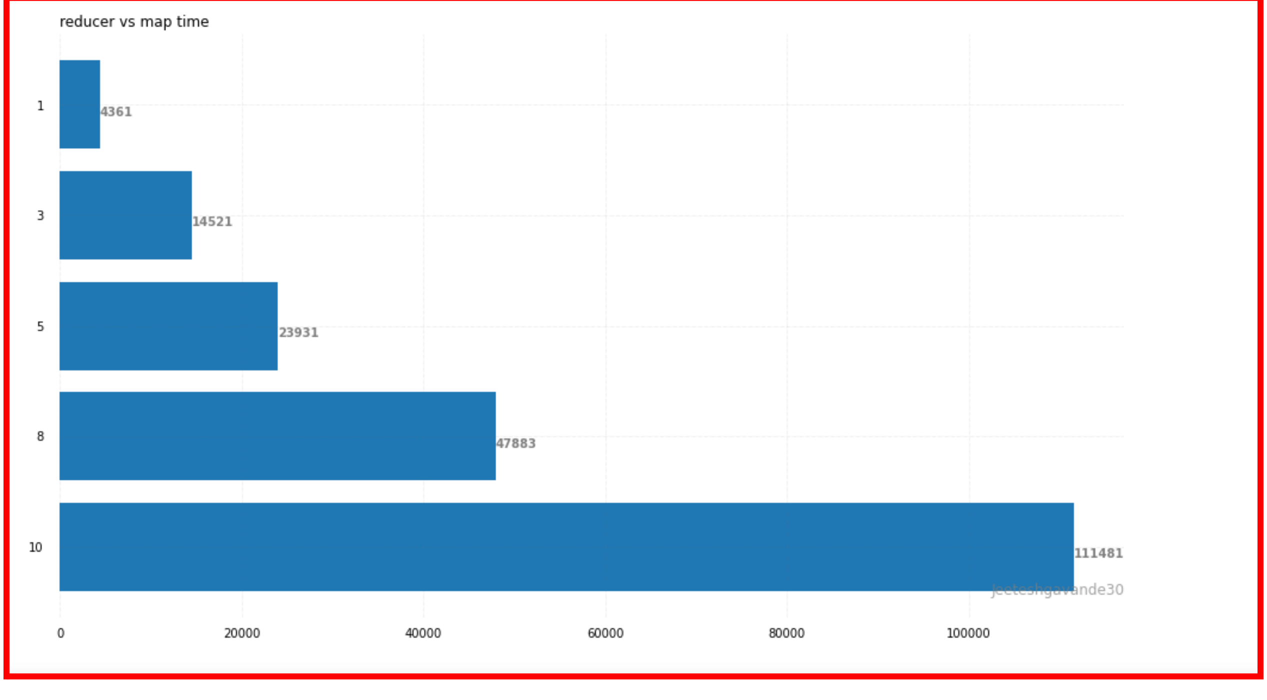
This programming approach is useful for assessing website and e-commerce platform usage trends. This framework can be used by companies that provide online services to strengthen their marketing tactics.

```
 Total vcore-milliseconds taken by all map tasks=7177
                Total vcore-milliseconds taken by all reduce tasks=14521
```

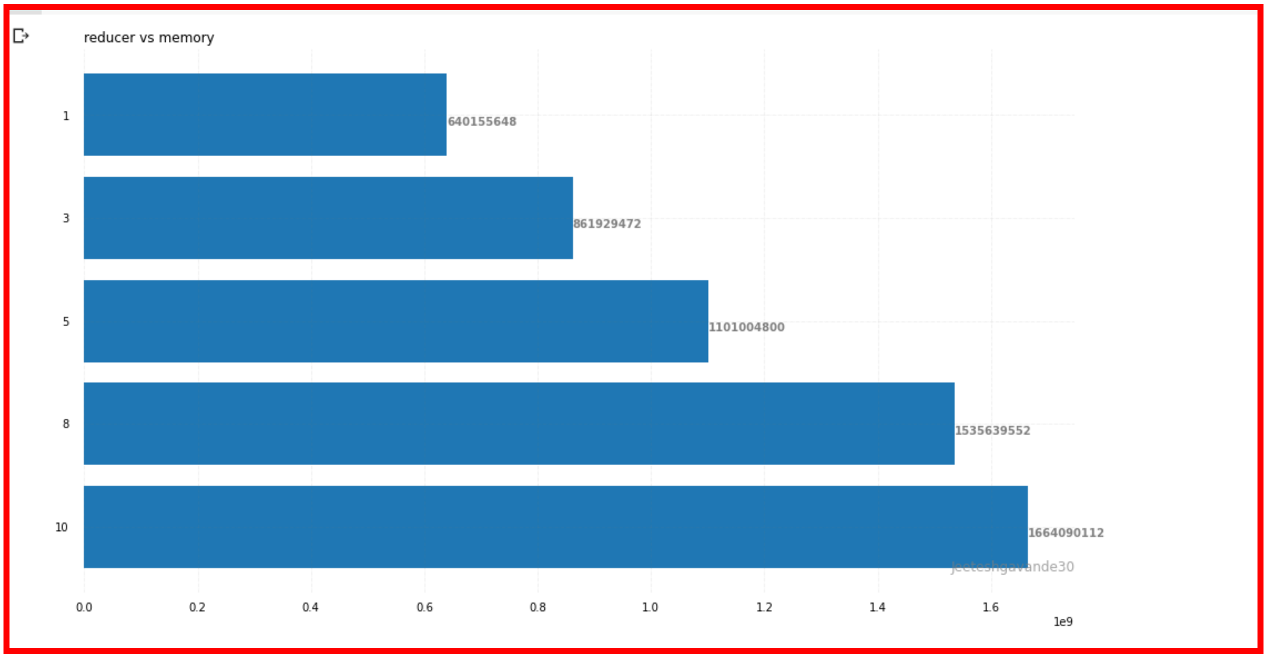
Reducer	Map time	Reduce time	Total committed heap usage (bytes)
1	9785	4361	640155648
3	7177	14521	861929472
5	6581	23931	1101004800
8	6225	47883	1535639552
10	13324	111481	1664090112

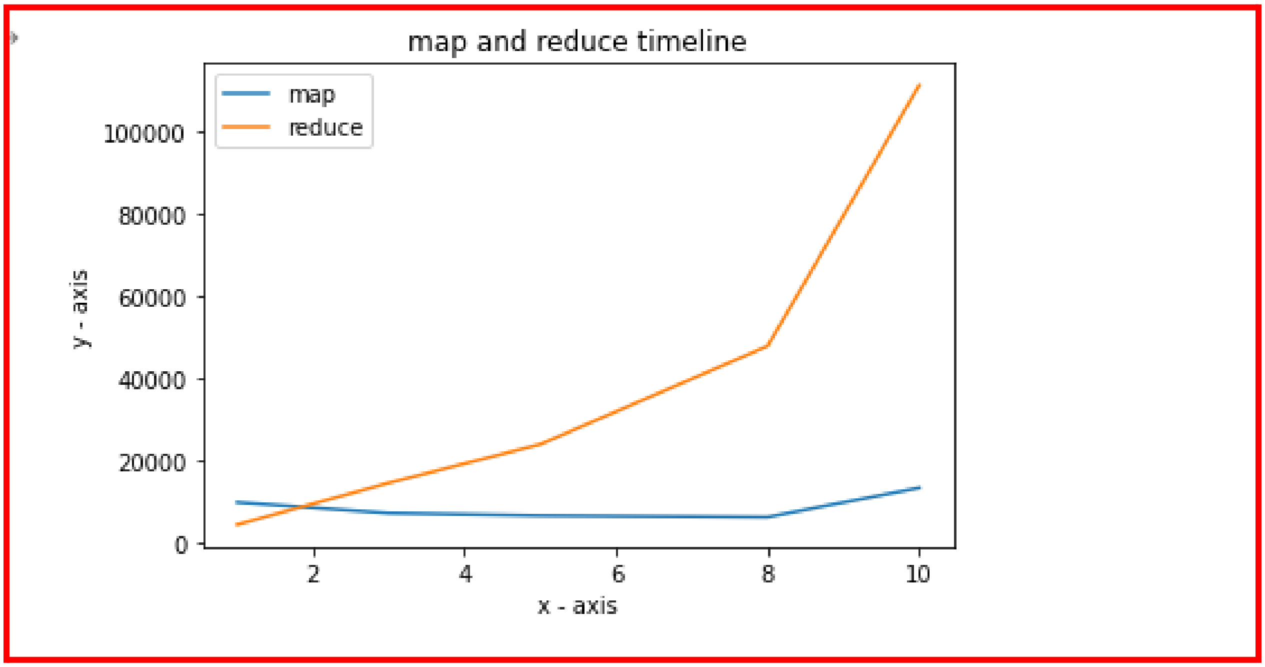


reducer vs map time



reducer vs map time


reducer vs memory
1
3
5
8
10
640155648
861929472
1101004800
1535639552
1664090112
Jeeteshgavande30
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e9


map and reduce timeline
map
reduce
y - axis
0
20000
40000
60000
80000
100000
2
4
6
8
10
x - axis