Name: Meet Vipul Gandhi
PRN No: 2020BTECS00112

High Performance Computing Lab

Practical No. 9

Title of practical: Implementation of Matrix-matrix Multiplication (global and shared Memory), Prefix sum, 2D Convolution using CUDA C

Problem Statement 1:
Implement Matrix-matrix Multiplication using global memory in CUDA C. Analyze and tune the program for getting maximum speed up. Do Profiling and state what part of the code takes the huge amount of time to execute.

Screenshots:

a. <<<1,64>>>

```
In [29]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  --------------------
    99.1        206484670          3  68828223.3     3577  206389364  cudaMalloc
     0.8          1612585          3    537528.3   273152     944231  cudaMemcpy
     0.2           333421          1    333421.0   333421     333421  cudaDeviceSynchronize
     0.0            33891          1     33891.0    33891      33891  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum              Name
 -------  ---------------  ---------  --------  -------  -------  ------------------------------
   100.0           191996          1  191996.0   191996   191996  multiply(int*, int*, int*, int)
```

b. <<<1,128>>>

```
In [31]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  --------------------
    99.1        228128248          3  76042749.3     5527  228006628  cudaMalloc
     0.8          1740113          3    580037.7   264514    1046168  cudaMemcpy
     0.1           322146          1    322146.0   322146     322146  cudaDeviceSynchronize
     0.0            43395          1     43395.0    43395      43395  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum              Name
 -------  ---------------  ---------  --------  -------  -------  ------------------------------
   100.0           188827          1  188827.0   188827   188827  multiply(int*, int*, int*, int)
```

c. <<<1,256>>>

```
In [33]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.2        252927473          3  84309157.7     5113  252799044  cudaMalloc
     0.6          1653679          3    551226.3   252674     999219  cudaMemcpy
     0.1           338457          1    338457.0   338457     338457  cudaDeviceSynchronize
     0.0            33058          1     33058.0    33058      33058  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum              Name
 -------  ---------------  ---------  --------  -------  -------  --------------------------------
   100.0           193403          1  193403.0   193403   193403  multiply(int*, int*, int*, int)
```

d. <<<2,64>>>

```
In [35]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.0        201670431          3  67223477.0     3674  201577030  cudaMalloc
     0.8          1567593          3    522531.0   236622     937081  cudaMemcpy
     0.2           340201          1    340201.0   340201     340201  cudaDeviceSynchronize
     0.0            32307          1     32307.0    32307      32307  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum              Name
 -------  ---------------  ---------  --------  -------  -------  --------------------------------
   100.0           192155          1  192155.0   192155   192155  multiply(int*, int*, int*, int)
```

e. <<<2,128>>>

```
In [36]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.2        247476301          3  82492100.3     5455  247353095  cudaMalloc
     0.7          1649609          3    549869.7   250592     996163  cudaMemcpy
     0.1           328323          1    328323.0   328323     328323  cudaDeviceSynchronize
     0.0            41406          1     41406.0    41406      41406  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum              Name
 -------  ---------------  ---------  --------  -------  -------  --------------------------------
   100.0           191900          1  191900.0   191900   191900  multiply(int*, int*, int*, int)
```

f. <<<4,64>>>

```
In [37]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

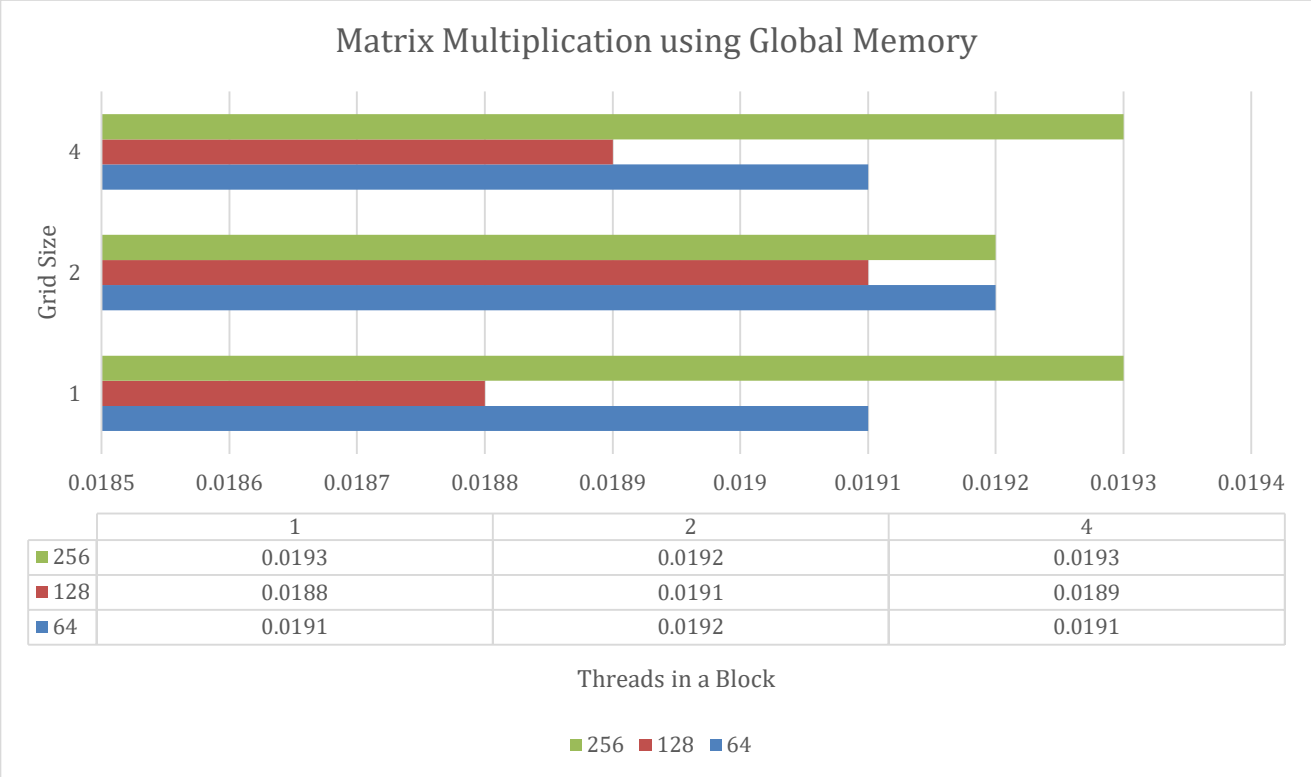
```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls    Average    Minimum   Maximum           Name
 -------  ---------------  ---------  -----------  -------  ---------  ---------------------
    99.3        374479339          3  124826446.3     7207  374302959  cudaMalloc
     0.6          2415696          3     805232.0   417017    1439206  cudaMemcpy
     0.1           297806          1     297806.0   297806     297806  cudaDeviceSynchronize
     0.0            64541          1      64541.0    64541      64541  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average   Minimum  Maximum             Name
 -------  ---------------  ---------  --------  -------  -------  --------------------------------
   100.0           191804          1  191804.0   191804   191804  multiply(int*, int*, int*, int)
```

Analysis:



	1	2	4
256	0.0193	0.0192	0.0193
128	0.0188	0.0191	0.0189
64	0.0191	0.0192	0.0191

Problem Statement 2:
Implement Matrix-matrix Multiplication using shared memory in CUDA C. Analyze and tune the program for getting maximum speed up. Do Profiling and state what part of the code takes the huge amount of time to execute.

Screenshots:

a. <<<1,64>>>

```
In [54]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.4        287791207          3  95930402.3     4818  287669265  cudaMalloc
     0.6          1650080          3    550026.7   250132     996875  cudaMemcpy
     0.1           149805          1    149805.0   149805     149805  cudaDeviceSynchronize
     0.0            40523          1     40523.0    40523      40523  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average  Minimum  Maximum              Name
 -------  ---------------  ---------  -------  -------  -------  --------------------------------
   100.0             8767          1   8767.0     8767     8767  multiply(int*, int*, int*, int)
```

b. <<<1,128>>>

```
In [56]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.2        215468038          3  71822679.3     3597  215371985  cudaMalloc
     0.7          1514654          3    504884.7   239053     880680  cudaMemcpy
     0.1           154061          1    154061.0   154061     154061  cudaDeviceSynchronize
     0.0            33506          1     33506.0    33506      33506  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average  Minimum  Maximum              Name
 -------  ---------------  ---------  -------  -------  -------  --------------------------------
   100.0             8896          1   8896.0     8896     8896  multiply(int*, int*, int*, int)
```

c. <<<1,256>>>

```
In [58]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.2        206345585          3  68781861.7     3647  206247785  cudaMalloc
     0.7          1542437          3    514145.7   257659     891819  cudaMemcpy
     0.1           164952          1    164952.0   164952     164952  cudaDeviceSynchronize
     0.0            26032          1     26032.0    26032      26032  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average  Minimum  Maximum              Name
 -------  ---------------  ---------  -------  -------  -------  --------------------------------
   100.0            10144          1  10144.0    10144    10144  multiply(int*, int*, int*, int)
```

d. <<<2,128>>>

```
In [60]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

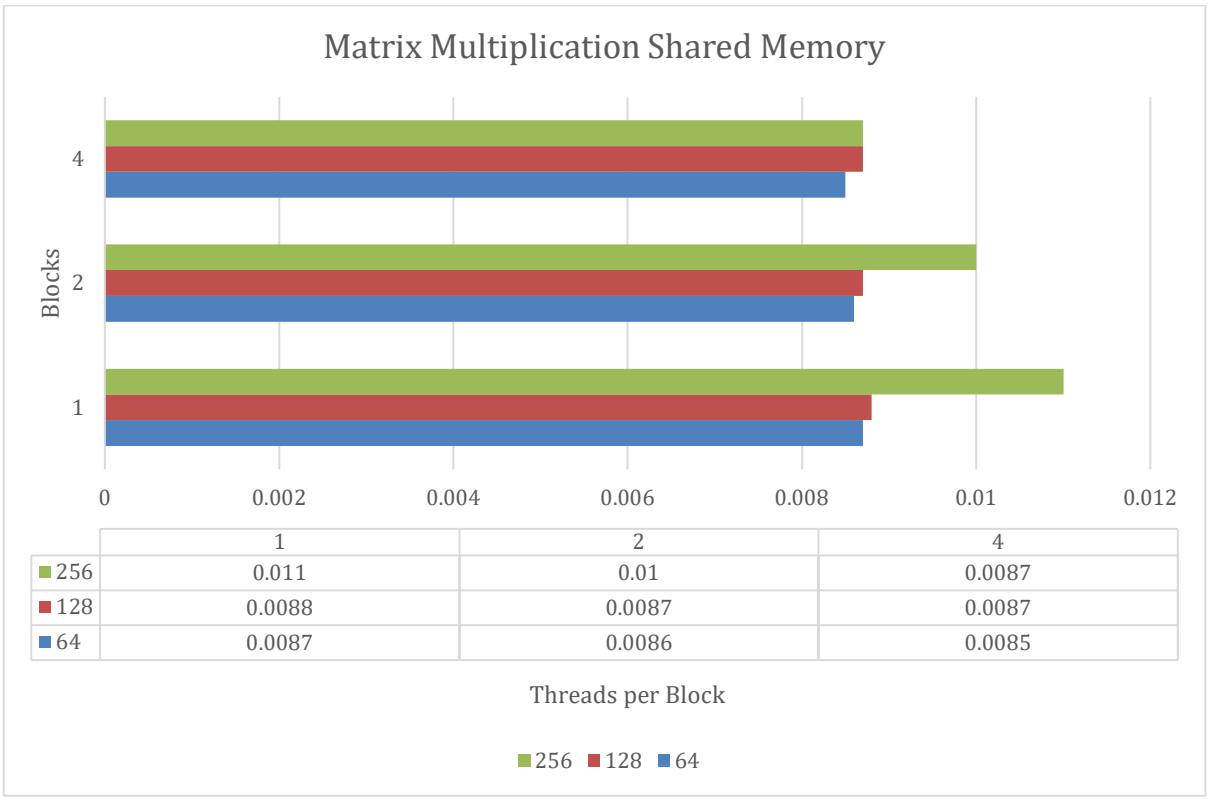
```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls   Average    Minimum   Maximum           Name
 -------  ---------------  ---------  ----------  -------  ---------  ---------------------
    99.2        219537952          3  73179317.3     3720  219440723  cudaMalloc
     0.7          1568912          3    522970.7   237260     937717  cudaMemcpy
     0.1           162660          1    162660.0   162660     162660  cudaDeviceSynchronize
     0.0            24492          1     24492.0    24492      24492  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances  Average  Minimum  Maximum              Name
 -------  ---------------  ---------  -------  -------  -------  ------------------------------
   100.0             8768          1   8768.0     8768     8768  multiply(int*, int*, int*, int)
```

Analysis:



	1	2	4
256	0.011	0.01	0.0087
128	0.0088	0.0087	0.0087
64	0.0087	0.0086	0.0085

Problem Statement 3:

Implement Prefix sum using CUDA C. Analyze and tune the program for getting maximum speed up. Do Profiling and state what part of the code takes the huge amount of time to execute.

Screenshots:

a. Block Size: 64

```
In [96]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls    Average    Minimum    Maximum         Name
 -------  ---------------  ---------  -----------  --------  ---------  ----------------
    60.7        222291507          2  111145753.5    224684  222066823  cudaMalloc
    38.6        141163625          2   70581812.5  28117965  113045660  cudaMemcpy
     0.7          2635433          2    1317716.5    392528    2242905  cudaFree
     0.0            35624          1      35624.0     35624      35624  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances   Average    Minimum   Maximum                      Name
 -------  ---------------  ---------  ----------  --------  --------  ----------------------------------------
   100.0         14150833          1  14150833.0  14150833  14150833  prefixSumKernel(int*, int*, unsigned long)
```

b. Block Size: 128

```
In [100]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls    Average    Minimum    Maximum         Name
 -------  ---------------  ---------  -----------  --------  ---------  ----------------
    65.1        257028696          2  128514348.0    378160  256650536  cudaMalloc
    34.5        136448064          2   68224032.0  28288341  108159723  cudaMemcpy
     0.4          1576879          2     788439.5    399069    1177810  cudaFree
     0.0            34106          1      34106.0     34106      34106  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances   Average    Minimum  Maximum                      Name
 -------  ---------------  ---------  ----------  -------  -------  ----------------------------------------
   100.0          9353922          1  9353922.0  9353922  9353922  prefixSumKernel(int*, int*, unsigned long)
```

c. Block Size: 256

```
In [99]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls    Average    Minimum    Maximum         Name
 -------  ---------------  ---------  -----------  --------  ---------  ----------------
    63.2        238077197          2  119038598.5    240454  237836743  cudaMalloc
    36.1        135988854          2   67994427.0  27828081  108160773  cudaMemcpy
     0.7          2623357          2    1311678.5    384088    2239269  cudaFree
     0.0            26163          1      26163.0     26163      26163  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances   Average   Minimum  Maximum                    Name
 -------  ---------------  ---------  ---------  -------  -------  ----------------------------------------
   100.0          9353792          1  9353792.0  9353792  9353792  prefixSumKernel(int*, int*, unsigned long)
```

d. Block Size: 512

```
In [104]: !nsys profile --stats=true -o vector-add-no-prefetch-report ./vector-add-no-prefetch
```

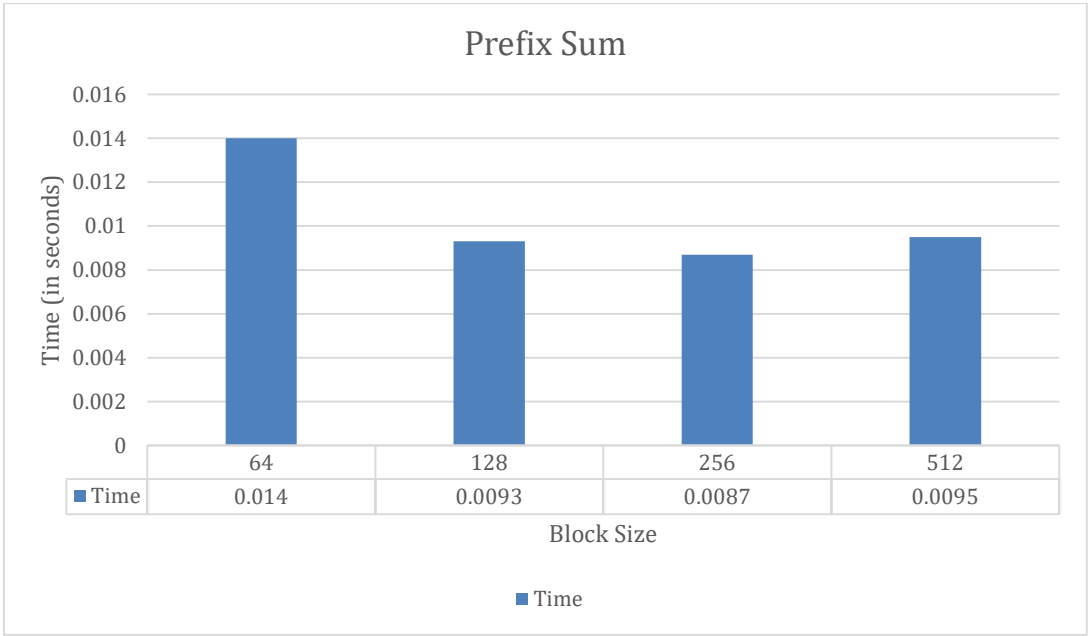
```
CUDA API Statistics:

 Time(%)  Total Time (ns)  Num Calls    Average    Minimum    Maximum         Name
 -------  ---------------  ---------  -----------  --------  ---------  ----------------
    61.6        222409847          2  111204923.5    221378  222188469  cudaMalloc
    38.0        137170879          2   68585439.5  28100197  109070682  cudaMemcpy
     0.4          1573325          2     786662.5    364068    1209257  cudaFree
     0.0            39405          1      39405.0     39405      39405  cudaLaunchKernel


CUDA Kernel Statistics:

 Time(%)  Total Time (ns)  Instances   Average   Minimum  Maximum                    Name
 -------  ---------------  ---------  ---------  -------  -------  ----------------------------------------
   100.0          9504251          1  9504251.0  9504251  9504251  prefixSumKernel(int*, int*, unsigned long)
```

Analysis:



Problem Statement 4:

Implement 2D Convolution using shared memory using CUDA C. Analyze and tune the program for getting maximum speed up. Do Profiling and state what part of the code takes the huge amount of time to execute.

Image size: 256*256

Mask Size: 7*7

Screenshots:

a. 64 Threads

```
In [116]: !nvcc -o vector-add-no-prefetch 01-vector-add/01-vector-add.cu -run

GPU Execution Time: 0.003808 ms
CPU Execution Time: 46.086000 ms
Speedup: 12102.416016
```

b. 128 Threads

```
In [117]: !nvcc -o vector-add-no-prefetch 01-vector-add/01-vector-add.cu -run

GPU Execution Time: 0.002912 ms
CPU Execution Time: 38.331000 ms
Speedup: 13163.118164
```

c. 256 Threads

```
In [118]: !nvcc -o vector-add-no-prefetch 01-vector-add/01-vector-add.cu -run

GPU Execution Time: 0.002592 ms
CPU Execution Time: 24.692000 ms
Speedup: 9526.234375
```

d. 512 Threads

```
In [119]: !nvcc -o vector-add-no-prefetch 01-vector-add/01-vector-add.cu -run

GPU Execution Time: 0.003328 ms
CPU Execution Time: 25.232000 ms
Speedup: 7581.730957
```

GitHub: https://github.com/meetgandhi692/HPC-Lab/tree/7e35953168fb8ce9749ba75e07871908464ba1b1/Assignment%209