Assignment 6 - Strassen's multiplication on GPU

Ghazal Rafiei

May 16, 2022

1 Introduction

The objective in this assignment is to parallelize Strassen's algorithm using Cuda and exploiting GPUs in order to improve performance. In the rest of this document, we will discuss the platform specifications, implementation approach, experiment method and the results.

2 Platform

The program is written in C++ using Cuda. Furthermore, here is the specification of the system:

```
Linux zenbookux434flcux433flc 5.15.32-1-MANJARO 1 SMP PREEMPT x86_64 GNU/Linux
CPU(s):  8
Vendor ID: GenuineIntel
Model name:  Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz
MemTotal:  16179008 kB
GPU: NVIDIA GP108BM [GeForce MX250] driver:  nvidia v:  510.60.02
CUDA Driver Version / Runtime Version:  11.6 / 11.6
```

3 Parallel Implementation

To parallelize Strassen's algorithm, we use GPU to calculate arbitrary summation and subtraction of submatrices in the algorithm.

4 Experiment

Multiplication of matrices with sizes 256, 512, 1024, 2048 and 4096 is calculated 10 times with GPU with block size 32, and on CPU. Each reported duration is average of 10 times of execution.

5 Results

In the following table, we can see the results of explained experiment in the previous section.

	Matrix Size				
Strassen algorithm (ad-hoc=256)	**256**	**512**	**1024**	**2048**	**4096**
Parallel on GPU	0.0009	0.003	0.025	0.16	0.83
Serial	0.20	1.14	9.17	66.68	481.40
Speed Up	222.2	380.0	366.8	416.8	580.0

Table 5.1 - Comparison duration of matrix multiplication between GPU and CPU.

As it is shown in the table 5.1, GPU provides hundreds of times speedup for this particular algorithm compared to its serial mode on CPU. The speedup rate of this method compared to CPU is 222, 380, 366, 416 and 580 respectively. As you can observe, the speedup rate increases as the size of matrices increase.

6 Conclusion

To conclude, GPU can make Strassen's algorithm hundreds of times faster. Moreover, The bigger the matrix becomes, the faster will GPU compute.