Programowanie Równoległe na Architekturach Wielordzeniowych

Parallel programming for multi-core architectures

2023/2024

Parallel implementation evaluation report

1. Project details

Project title:	Matrix Multiplication
Student's name:	Piotr Wróblewski
Index number:	181655
Email:	S181655@poczta.pg.edu.pl

2. Description of data used for experiments (including examples, when possible)

For all experiments I used randomly generated matrices, of sizes <100, 1000; step 100>.
Random generator implementation:

```
std::default_random_engine generator;
std::uniform_real_distribution<float> distribution(0.0, 1.0);
for (size_t i = 0; i < N * N; i++)
{
    A[i] = distribution(generator);
    B[i] = distribution(generator);
}
```

3. Environment #1 description

System: Ubuntu 22.04.3
CPU: Intel Xeon 4210
GPU: RTX 5000
CUDA: 12.2
OpenMP: 11.4.0

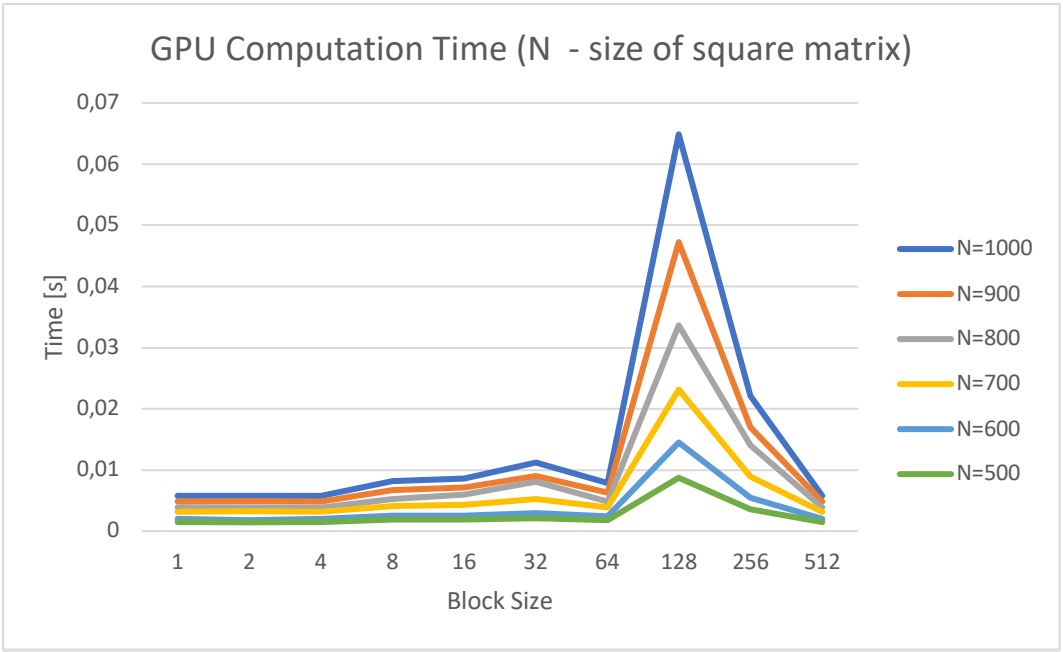
4. Test results

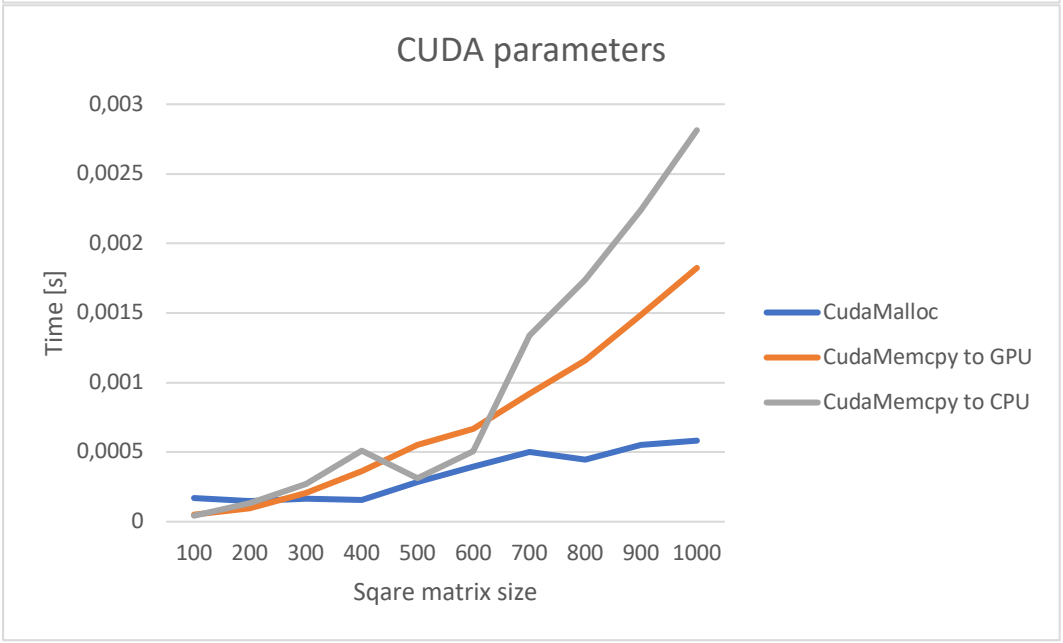
Implementation	**Execution time***	
	Mean [s]	**Uncertainty [s]**
OpenMP, 4 threads, N=500	0,09795072	0,00157748
CUDA, Block Size=4, N=500	0,002670977	0,00000957478

* calculated over 10 executions, uncertainty calculated as:

N	OpenMP	GPU
500	0,1005	0,00267359
500	0,0969784	0,00267322
500	0,0964838	0,00269539
500	0,0972094	0,00267202
500	0,0982789	0,00266515
500	0,0960671	0,00266506
500	0,0997854	0,00265589
500	0,0962112	0,00266763
500	0,0990844	0,00267079
500	0,0989086	0,00267103
mean:	0,09795072	0,002670977
uncertainty:	0,00157748	9,57478E-06

5. Implementation CUDA, OpenMP






GPU Kernel and GPU total computation time comparasion
0,007
0,006
0,005
0,004
0,003
0,002
0,001
0
Time [s]
100
200
300
400
500
600
700
800
900
1000
Square matrix size
GPU Duration
Kernel Duration
OpenMP Computation Time (N - size of square matrix)
0,6
0,5
0,4
0,3
0,2
0,1
0
Time [s]
2
3
4
5
6
7
8
9
10
11
Number of threads
N=1000
N=900
N=800
N=700
N=600
N=500
Performance comparasion
0,6
0,5
0,4
0,3
0,2
0,1
0
Time [s]
100
200
300
400
500
Square matrix size
GPU Duration (Block size =4)
OpenMP (Num. threads = 6)
Single Core

6. Survey:

   Fill the answers for questions related to frameworks that you used in your project:

   a. How many lines of code did you write for:

      i. OpenMP implementation: ……40…….

      ii. CUDA/OpenCL implementation: ……70…….

   b. How would you describe programming difficulty of each framework/interface in 1-10 scale (1 – easy, 10 – difficult):

      i. OpenMP: ……3….

      ii. CUDA: ……7….