# Multi-core Architectures 2021/2022

# Parallel implementation evaluation report

1. Project details

|  |  |
| --- | --- |
| Project title: | Matrix multiplication |
| Student’s name: | Emil Walewski |
| Index number: | 188152 |
| Email: | s188152@student.pg.edu.pl |

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

|  |
| --- |
| Auto generated data  void fillMatrixRandomValues(float\* matrix, int rows, int columns){      for(int i = 0; i < rows; i++){          for(int j = 0; j < columns; j++){              matrix[i \* columns + j] = rand() % 9 + 1;          }      }  } |

1. Environment #1 description

|  |
| --- |
| **Operating system:** Linux des16.kask.eti.pg.gda.pl 5.4.0-81-generic #91-Ubuntu SMP Thu Jul 15 19:09:17 UTC 2021 x86\_64 x86\_64 x86\_64 GNU/Linux  **CPU:** Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz  **GPU:** GP106 [GeForce GTX 1060 6GB]  **Library version:** cuda 11 |

1. Environment #2 description

|  |
| --- |
| *Operating system, CPU/GPU model, libraries versions (CUDA, OpenMP, MPI, etc.)* |

1. Test results

|  |  |  |
| --- | --- | --- |
| **Implementation** | **Execution time\*** | |
| **Mean [s]** | **Uncertainty [s]** |
| <Impl. #1 name> (e.g. OpenMP-based) |  |  |
| <Impl. #2 name> (e.g. CUDA-based) |  |  |

\* calculated over 10 executions, uncertainty calculated as:

1. Implementation #1 details

|  |
| --- |
|  |

1. Implementation #2 details

|  |
| --- |
|  |

1. Survey

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

* 1. How many lines of code did you write for:
     1. OpenMP implementation: …………….
     2. CUDA/OpenCL implementation: …………….
  2. How would you describe programming difficulty of each framework/interface in 1-10 scale (1 – easy, 10 – difficult):
     1. OpenMP: …………….
     2. CUDA/OpenCL: …………….