**Laboratory exercises**

**Problem: Matrix multiplication**

A (n x k) x B (k x m) =C (n x m)

Elements are real numbers (double).

n= 1024; m = 1024

Matrices should be read from an input file (the file should be previously generated with random numbers).

The result is written into an output file.

**L1**) sequential

**L2)** multithreading – explicit threads creation and management ( t threads- t is given as parameter)

Two variants:

**L2a)** Row decomposition (n/p \*m elements assigned to each thread)

**L2b)** Block decomposition p=p1\*p1 => n/p1\*m/p2 elements assigned to each thread

Correctness verification => compare the result with that obtained at L1!

Testing for:

t= 4, 8,16, 32 threads

**L3)** multithreading - OpenMP ( t threads- t is given as parameter) - optional

t= 4,8,16

Correctness verification => compare the result with that obtained with L1!

**L4)** multiprocessing – MPI ( p processes – p is given as parameter)

p= 4,8,16,32

Cannon algorithm!!! (https://iq.opengenus.org/cannon-algorithm-distributed-matrix-multiplication/)

Correctness verification => compare the result with that obtained at L1!

**L5)** hybrid multiprocessing + multithreading ( p processes – each process uses t threads- p and t are given as parameters)

Cannon algorithm +multithreaded local parallelization (inside a process)

(p,t)= (4,4), (8,4),(16,4), (4,16)

Correctness verification => compare the result with that obtained at L!

Documentation should be provided for each variant, and it should include the design and the performance analysis.

All the performance results should be documented, and each test should be repeated 10 times (average time will be considered).

-Performance analysis:

-execute each variant 10 times with the same input data

- measure the execution time for each execution

-compute the average of the execution time of all 10 execution

(execution scripts could be used)

**Table 1: Total execution time**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | T1 | T2 | T3 | T4 | Speed-up S=(Ts/Tp) Ts=L1-T1 | Efficiency (E=S/p) | Cost (C= Tp\*p) |
| L1 |  |  |  |  |  |  |  |
| L2a |  |  |  |  |  |  |  |
| L2b |  |  |  |  |  |  |  |
| L3 |  |  |  |  |  |  |  |
| L4 |  |  |  |  |  |  |  |
| L5 |  |  |  |  |  |  |  |

**Table 2: Computation time without reading and writing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | T1 | T2 | T3 | T4 | Speed-up S=(Ts/Tp) Ts=L1-T1 | Efficiency (E=S/p) | Cost (C= Tp\*p) |
| L1 | 4.84453 |  |  |  |  |  |  |
| L2a | 1.59272 | 1.76967 | 2.20765 | 1.92743 |  |  |  |
| L2b | 2.02139 | 2.03394 | 1.96328 | 1.9734 |  |  |  |
| L3 |  |  |  |  |  |  |  |
| L4 |  |  |  |  |  |  |  |
| L5 |  |  |  |  |  |  |  |

Times should be measured on the same architecture in the same conditions – no other workload.

**Deadlines**

|  |  |  |  |
| --- | --- | --- | --- |
| Deadlines | Grade10 | Grade 8 | Grade 6 |
| Seminar 2 | L1, L2 |  |  |
| Seminar 3 | L2, L3 | L1 |  |
| Seminar 4 | L4 | L2 | L1 |
| Seminar 5 | L5 | L4 | L2 |
| Seminar 6 |  | L5 | L4 |

At one seminar a maximum of 2 labs could be presented.