Documentatie proiect

-Inmultire de matrici C++, MPI-

Teisanu Mihai, CR 3.3B

I.Varianta secventiala C++.

Algoritm: Inmultire clasica linie cu coloana O(n^3).

Input: 10 inputuri - 2 matrici cu elemente random pana la 100 de dimensiuni 8, 16, 32, ..., 4096.

Output: timpul mediu de rulare pentru fiecare input si in alt fisier matricea rezultat

Mod rulare: Matricile din input sunt patratice. Dimensiuninea si elementele se citesc din fisier. Inmultirea se executa de un numar de ori (5 pentru testele facute) si se calculeaza timpul mediu de rulare - doar pentru inmultire nu si pentru citirea/scrierea datelor-.

Corectitudinea algoritmului: Am verificat facand inmultirile si in Matlab. Pentru primul exemplu in0.txt se observa ca outputul din Matlab e acelasi cu cel din out0.txt.

Command Window

```
>> a

a =

    38    70    55    25    56    45    17    23
    14    70    71    85    37    39    76    58
    63     8    47    44    78    65    52    27
    39    68     5     5    29    92    21    34
    30    71    91    30    10    84    90    71
    79    68    24    87    88    22    60    30
    73    70    77    91    62    36     4    65
    32    32    76   100    24    11    20    29

>> b

b =

    77     7    88    10    44    92    38    65
    89    60    72    20    53    72    29    97
    34    47    80    11    87    86    37    84
     5    81    47    40    65    30    52    67
    78    39    87    78    49    12    52    44
    15    25    63    22    24    65    82    79
    78    60     8     2    39    82    54     8
    47    40    97    75    55    26    40    99

>> c = a * b

c =

       18601       14325       24033       10502       17544       19605       15249       23987
       22272       23818       27857       13967       24931       25703       21155       30418
       19765       15561       25864       12710       19040       21861       19927       23423
       16128       11044       20748        8883       12398       17998       15445       21964
       24270       21907       30211       12054       24871       31404       23234       33316
       26670       21590       30289       15612       23653       25024       21206       28636
       23667       21859       35900       17128       27169       26210       21377       36139
       13356       17387       21654       10125       20031       18181       14546       23224
```

main.cpp M out0.txt M

Outputs > out0.txt

```
1
2  18601 14325 24033 10502 17544 19605 15249 23987
3  22272 23818 27857 13967 24931 25703 21155 30418
4  19765 15561 25864 12710 19040 21861 19927 23423
5  16128 11044 20748 8883 12398 17998 15445 21964
6  24270 21907 30211 12054 24871 31404 23234 33316
7  26670 21590 30289 15612 23653 25024 21206 28636
8  23667 21859 35900 17128 27169 26210 21377 36139
9  13356 17387 21654 10125 20031 18181 14546 23224
10
```

Specificatiile calculatorului pe care s-au rulat testele:

View basic information about your computer

Windows edition

Windows 11 Pro

© Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
Installed memory (RAM):	32.0 GB (31.8 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Specificatii CPU:

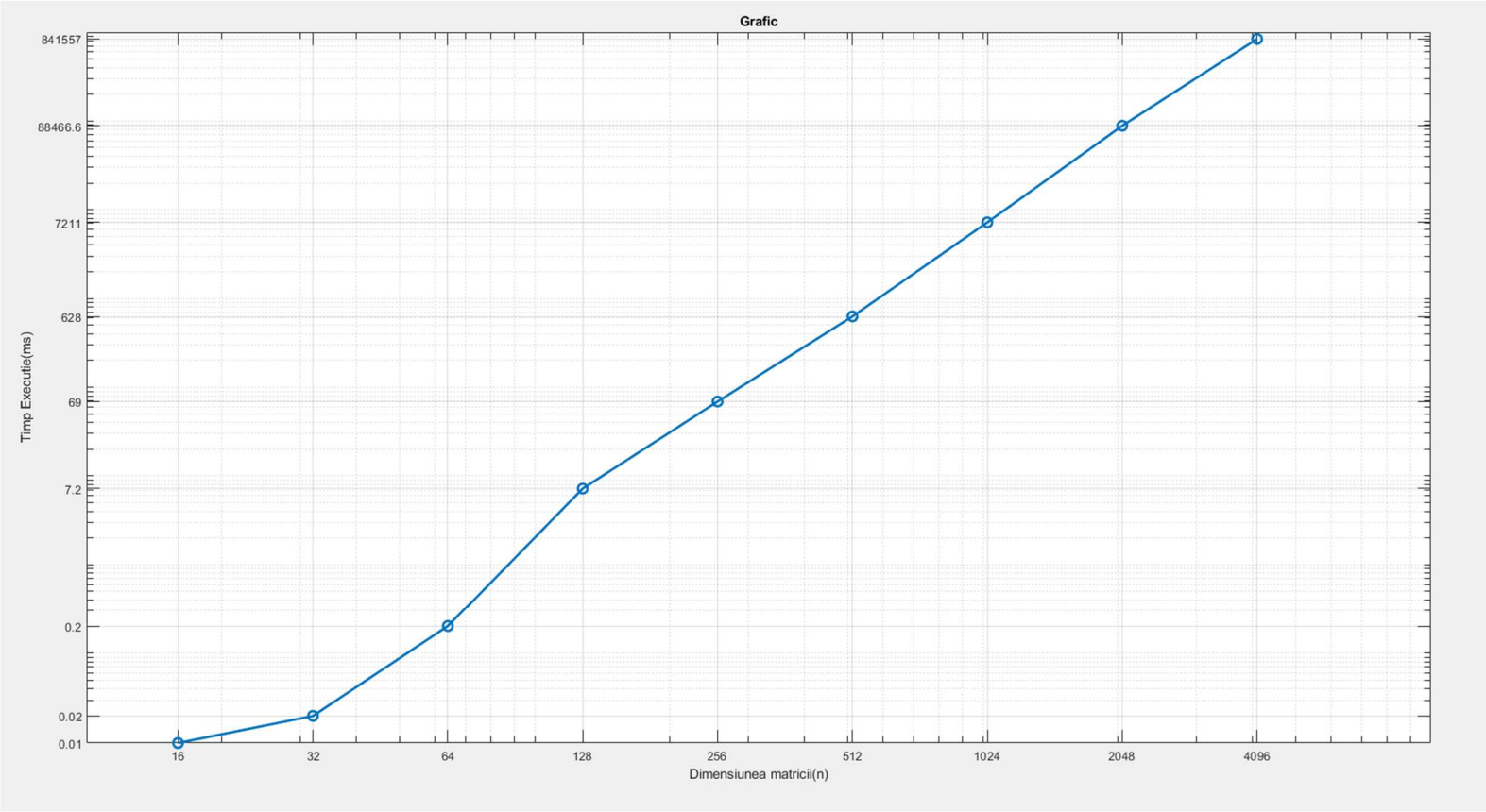
Utilization	Speed	Processes	Threads	Handles
5%	4.56 GHz	283	4194	124662

Up time

0:03:45:48

Base speed:	2.59 GHz
Sockets:	1
Cores:	6
Logical processors:	12
Virtualization:	Enabled
L1 cache:	384 KB
L2 cache:	1.5 MB
L3 cache:	12.0 MB

Timpii medii de executie: Scala dublu logaritmica pentru dimensiunea matricelor de intrare si timpul mediu in ms:



Observatie: Avand O(n^3) complexitate si dubland dimensiunea matricelor la fiecare test ar trebui in principiu ca timpul pentru testul i + 1 sa fie de maxim 2^3 = 8 ori mai mare decat timpul pentru testul i. In rezultate timpul poate fi si de 12 ori mai mare intrucat datele pe care se lucreaza cresc (de 4 ori) si trebuie aduse mai des din memorie, cache-ul fiind limitat si umplandu-se mai des.