

# Tema 6

Bancescu Rares-Emil

## 1. Introducere: caracteristicile calculatorului personal

### a) HT ( Hyperthreaded systems ) API:

Rezultatele au fost obtinute in urma apelarii functiei GetLogicalProcessorInformation, functie cu care obtinem o structura de tipul SYSTEM\_LOGICAL\_PROCESSOR\_INFORMATION, ce are urmatoarea forma:

```
typedef struct _SYSTEM_LOGICAL_PROCESSOR_INFORMATION {
    ULONG_PTR ProcessorMask;
    LOGICAL_PROCESSOR_RELATIONSHIP Relationship;
    union {
        struct {
            BYTE Flags;
        } ProcessorCore;
        struct {
            DWORD NodeNumber;
        } NumaNode;
        CACHE_DESCRIPTOR Cache;
        ULONGLONG Reserved[2];
    } DUMMYUNIONNAME;
} SYSTEM_LOGICAL_PROCESSOR_INFORMATION, *PSYSTEM_LOGICAL_PROCESSOR_INFORMATION;
```

In continuare am afisat informatiile din campurile ProcessorMask si Relationship:

Numa nodes:1

Processor cores: 4

Cache 11 number: 8 size: 393216

Cache 12 number: 4 size: 2097152

Cache 13 number: 1 size: 4194304

ProcessorPackage: 1  
HT logical processor 3  
HT logical processor 12  
HT logical processor 48  
HT logical processor 192

**b) NUMA API:**

Pentru a obtine mastile de afinitate lpProcessAffinityMask si lpSystemAffinityMask am utilizat functiile GetProcessAffinityMask si GetNumaProcessorNode. Pentru numarul total de noduri am utilizat GetNumaHighestNodeNumber.

Am obtinut urmatoarele rezultate:

lpProcessAffinityMask: 255  
lpSystemAffinityMask: 255  
Total number of nodes: 1

**c) CPU Sets API:**

**1. Get processDefaultCpuSets:**

Pentru a obtine elementele din vectorul CpuSetIds am folosit functia GetProcessDefaultCpuSets, functie ce returneaza un vector ULONG CpuSetIds si lungimea acestui vector.

Am obtinut un vector cu 0 elemente:

Number of cpu sets:

## 2. GetSystemCpuSetInformation:

Pentru a obtine structurile SYSTEM\_CPU\_SET\_INFORMATION am utilizat functia GetSystemCpuSetInformation. Functia returneaza un vector de structuri SYSTEM\_CPU\_SET\_INFORMATION si lungimea acestuia. Structura este urmatoarea:

```
typedef struct _SYSTEM_CPU_SET_INFORMATION {
    DWORD          Size;
    CPU_SET_INFORMATION_TYPE Type;
    union {
        struct {
            DWORD    Id;
            WORD     Group;
            BYTE     LogicalProcessorIndex;
            BYTE     CoreIndex;
            BYTE     LastLevelCacheIndex;
            BYTE     NumaNodeIndex;
            BYTE     EfficiencyClass;
            union {
                BYTE AllFlags;
                struct {
                    BYTE Parked : 1;
                    BYTE Allocated : 1;
                    BYTE AllocatedToTargetProcess : 1;
                    BYTE RealTime : 1;
                    BYTE ReservedFlags : 4;
                } DUMMYSTRUCTNAME;
            } DUMMYUNIONNAME2;
            union {
                DWORD Reserved;
                BYTE SchedulingClass;
            };
            DWORD64 AllocationTag;
        } CpuSet;
    } DUMMYUNIONNAME;
} SYSTEM_CPU_SET_INFORMATION, *PSYSTEM_CPU_SET_INFORMATION;
```

Am obtinut un numar total de 8 Cpu sets, ale caror valori sunt urmatoarele:

The number of Current Process default CPU sets: 8

Processor set Id: 256

Processor set Group: 0

Processor set Logical Processor index: 0

Processor set core index: 0

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 257

Processor set Group: 0

Processor set Logical Processor index: 1

Processor set core index: 1

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 258

Processor set Group: 0

Processor set Logical Processor index: 2

Processor set core index: 2

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 259

Processor set Group: 0

Processor set Logical Processor index: 3

Processor set core index: 3

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 260

Processor set Group: 0

Processor set Logical Processor index: 4

Processor set core index: 4

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 261

Processor set Group: 0

Processor set Logical Processor index: 5

Processor set core index: 5

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 262

Processor set Group: 0

Processor set Logical Processor index: 6

Processor set core index: 6

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

Processor set Id: 263

Processor set Group: 0

Processor set Logical Processor index: 7

Processor set core index: 7

Processor set last level cache index: 0

Processor set numa node index: 0

Processor set efficiency class: 0

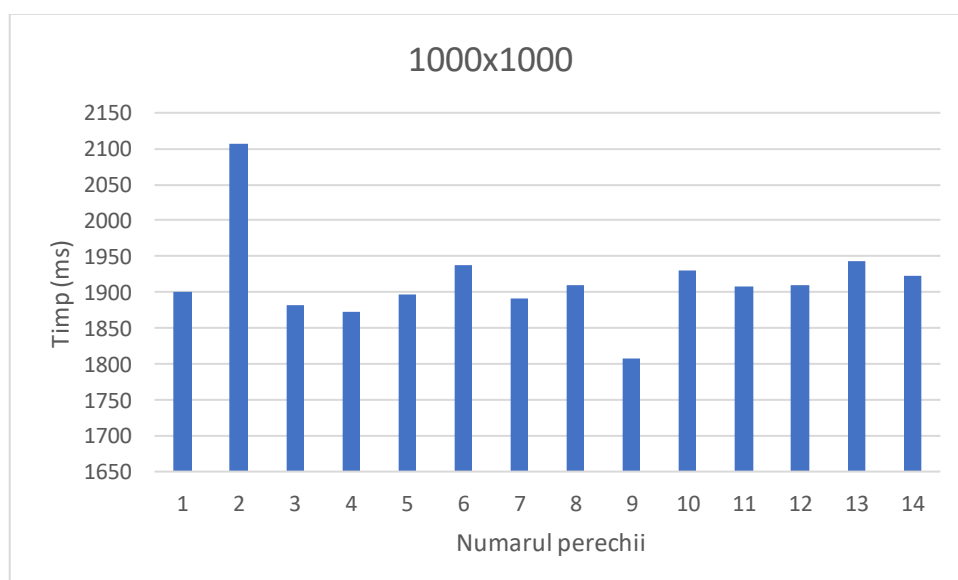
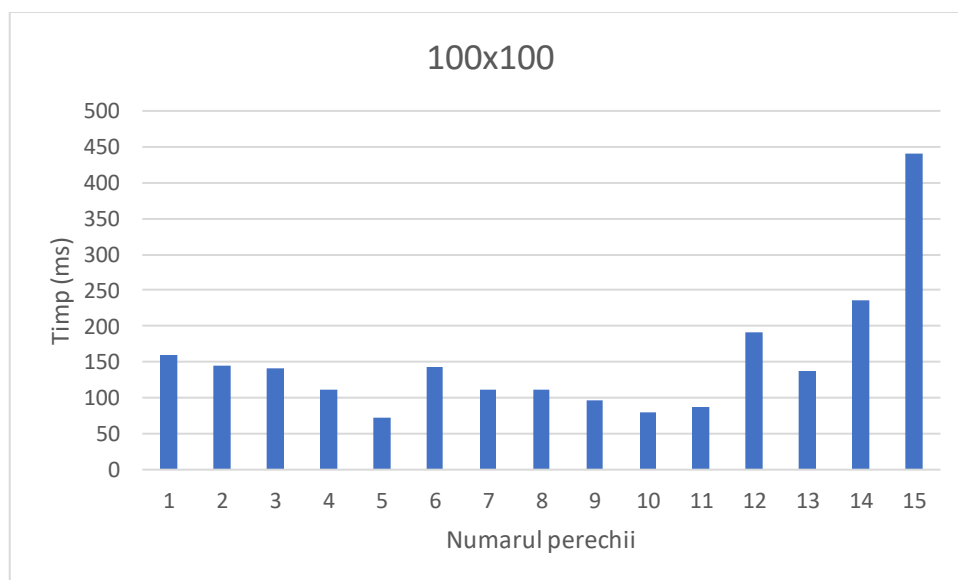
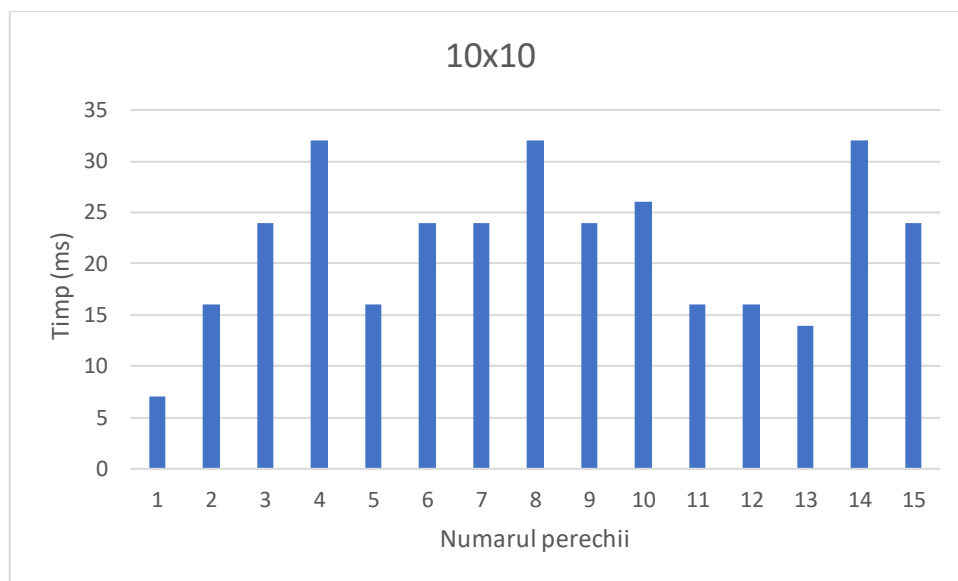
## **2. Evaluarea performantelor calculatorului personal:**

Pentru a evalua performantele calculatorului personal am efectuat o suită de calcule aritmetice ( Produsul a doua matrice ) atat in mod secvential, cat si in mod paralelizat static. Mediul de lucru pe care am rulat algoritmul de inmultire a matricelor este format din 3 seturi de cate 15 perechi de matrice A si B: Primul set este format din matrici de dimensiune 10x10, al doilea set este format din matrice de dimensiune 100x100, iar al treilea set este format din matrice de dimensiuni 1000x1000. Calculul secvential efectueaza inmultirea matricelor din fiecare pereche pas cu pas, de catre un singur thread. Calculul paralelizat imparte volumul de calcul al produsului celor 2 matrici dintr-o pereche in urmatorul mod celor 8 threaduri( Procesorul pe care rulez acest program are 4 nuclee fizice ):  $n/10 \cdot 2$  pentru primele doua threaduri si  $n/10$  pentru urmatoarele 6 threaduri, unde n este numarul de pasi pe care ii efectueaza bucla principala necesara inmultirii matricelor.

In urma rularii programului s-au obtinut urmatoarele rezultate:

- **Generarea mediul de lucru:**

Pentru generarea fiecarui set de cate 15 perechi de matrici s-au obtinut urmatoarii timpi:

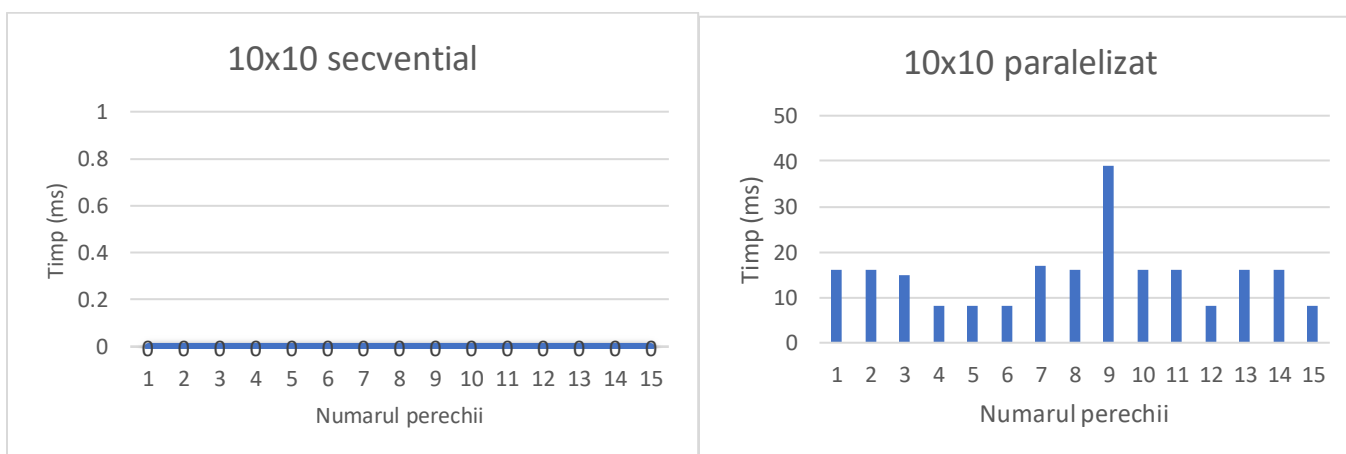


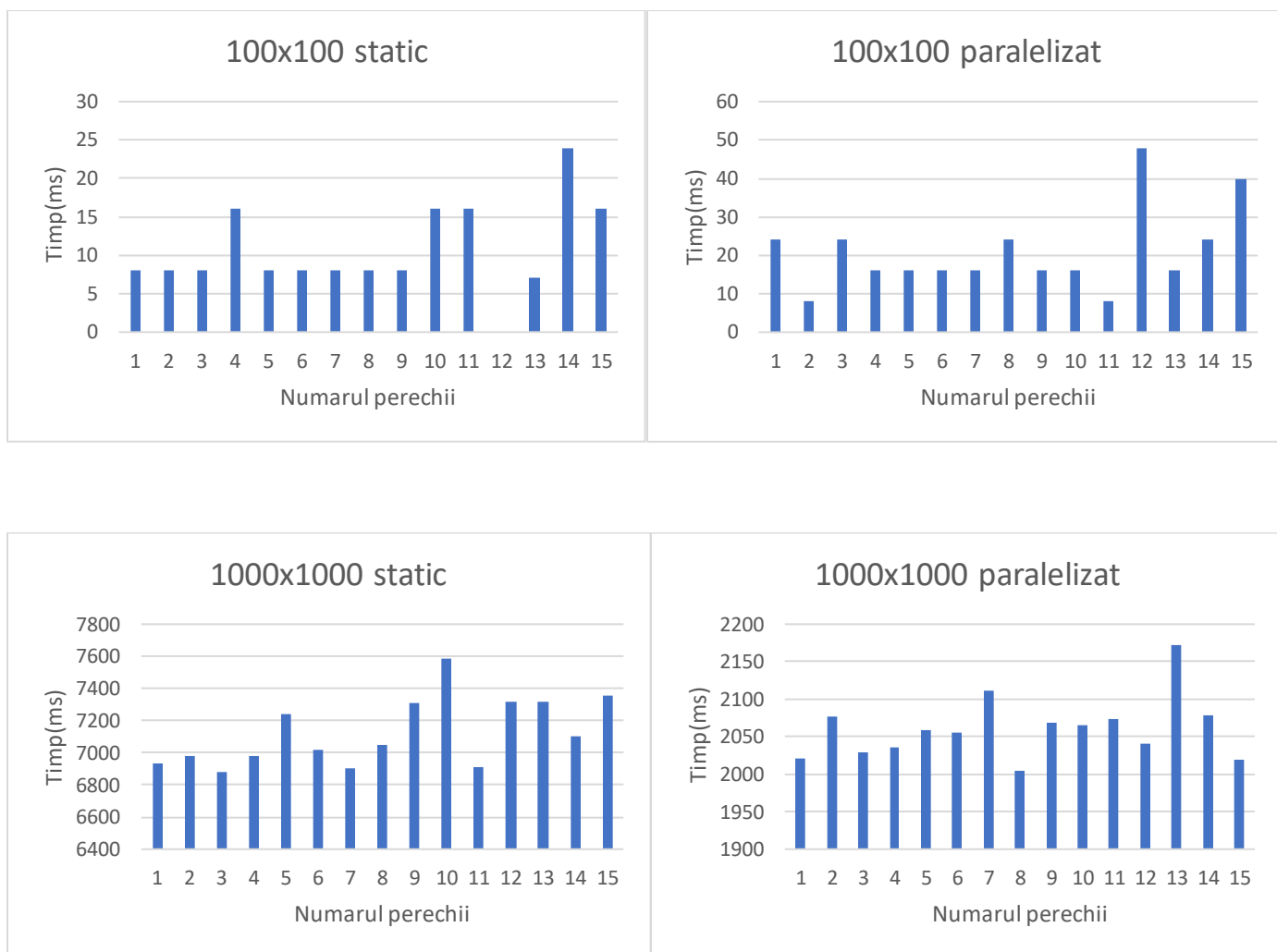


Mediile timpilor necesari generarii celor 15 perechi de matrice:

| Dimensiune: | 10x10  | 100x100  | 1000x1000 |
|-------------|--------|----------|-----------|
| Timp:       | 21.8ms | 151.26ms | 2002.86ms |

- Rezultatele obtinute:**





Mediile timpilor necesari inmultirii celor 15 perechi de matrice:

| Dimensiune        | 10x10    | 100x100 | 1000x1000  |
|-------------------|----------|---------|------------|
| Medie secvential  | 0 ms     | 10.6 ms | 7125.86 ms |
| Medie paralelizat | 14.86 ms | 20.8 ms | 2060.73 ms |

- **Concluzii:**

### **Generarea matricelor:**

Analizand media timpilor de generare putem observa ca timpul mediu al generarii unei perechi de matrice de dimensiune din ce in ce mai mare creste foarte mult. Principala cauza al acestui fapt este dimensiunea.

### **Timpii obtinuti la inmultirile matricelor:**

Analizand timpii obtinuti prin cele doua moduri de calcul secvential si paralelizat ajungem la urmatoarele concluzii:

1. Pentru dimensiuni mici, calculul in mod secvential este mai eficient. (matricele de dimensiuni  $10 \times 10$  si  $100 \times 100$ ). Principala cauza al acestui rezultat este ineficienta impartirii unui volum de calcul redus mai multor threaduri. Impartirea volumului de calcul si crearea fiecarui thread necesita timp suplimentar, timp in care calculul putea fi realizat.
2. Pentru dimensiuni mari ( $1000 \times 1000 \dots$ ) calculul in mod paralelizat este mult mai efficient. Volumul de calcul fiind exponential mai mare, impartirea acestuia mai multor threaduri va aduce o imbunatatire semnificativa timpului mediu de calcul al inmultirii matricelor.

Concluzia finala pe care o obtinem in urma analizei timpilor de executie este:  
Cand volumul de calcul nu este semnificativ, folosim metoda de calcul secventiala, iar cand avem un volum mare de calcul vom folosi modul paralelizat de calcul.