

Laborator nr. 9

Calcul matriceal

1 Înmulțirea a două matrice

Formularea problemei: date fiind matricele pătractice $A_{n \times n} = (a_{i,j})_{i,j=0,1,\dots,n-1}$ și $B_{n \times n} = (b_{i,j})_{i,j=0,1,\dots,n-1}$, se cere să se calculeze matricea pătratică $C_{n \times n} = (c_{i,j})_{i,j=0,1,\dots,n-1}$, conform formulei:

$$c_{i,j} = \sum_{k=0}^{n-1} a_{i,k} \times b_{k,j}, (\forall) i, j = 0, 1, \dots, n-1.$$

Algoritmul 1 prezintă algoritmul lui Cannon pentru înmulțirea a două matrice.

• **Notății:**

- P este o plasă formată din $q \times q = \sqrt{p} \times \sqrt{p} = p$ unități de procesare.
- $p_{i,j}$ este unitatea de procesare care execută algoritmul de înmulțire de blocuri (vezi curs 6 – algoritmul secvențial standard de înmulțire a două matrice).
- M_1 este blocul care urmează a fi transmis de $p_{i,j}$ către $p_{i,j \ominus 1}$.
- M_2 este blocul care urmează a fi transmis de $p_{i,j}$ către $p_{i \ominus 1, j}$.
- M_3 este blocul calculat de $p_{i,j}$.
- \oplus și \ominus semnifică adunarea și scăderea modulo q .

• **Premise:** Inițial, fiecare $p_{i,j}$ deține în memoria locală blocurile $A_{i,j}$ și $B_{i,j}$.

• **Rezultate:** Fiecare unitate de procesare $p_{i,j}$ calculează blocul $C_{i,j}$.

Algoritmul 1 Algoritmul lui Cannon

```
1: function INMULTIRE_MATRICE_CANNON( $A_{i,j}, B_{i,j}, n, q, p_{i,j}$ )
2:    $M_1 \leftarrow A_{i,j}; M_2 \leftarrow B_{i,j}$ 
3:   ALINIERE_INITIALA( $M_1, M_2, n, q, p_{i,j}$ )
4:    $M_3 \leftarrow 0$ 
5:   for  $k = 0$  to  $q - 2$  do
6:      $M_3 \leftarrow M_3 + \text{INMULTIRE\_MATRICE}(M_1, M_2, q)$ 
7:     trimite  $M_1$  către  $p_{i,j \ominus 1}$ 
8:     primește  $M_1$  de la  $p_{i,j \oplus 1}$ 
9:     trimite  $M_2$  către  $p_{i \ominus 1, j}$ 
10:    primește  $M_2$  de la  $p_{i \oplus 1, j}$ 
11:  end for
12:   $C_{i,j} \leftarrow M_3 + \text{INMULTIRE\_MATRICE}(M_1, M_2, q)$ 
13: end function
```

Figurile 1 – 3 prezintă un exemplu de rulare a Algoritmilor 1 și 2.

Algoritmul 2 Algoritmul lui Cannon – alinierea inițială a matricelor

```

1: function ALINIERE_INITIALA( $M_1, M_2, n, q, p_{i,j}$ )
2:   for  $k = 1$  to  $i$  do
3:     trimite  $M_1$  către  $p_{i,j \oplus 1}$ 
4:     primește  $M_1$  de la  $p_{i,j \oplus 1}$ 
5:   end for
6:   for  $k = 1$  to  $j$  do
7:     trimite  $M_2$  către  $p_{i \oplus 1, j}$ 
8:     primește  $M_2$  de la  $p_{i \oplus 1, j}$ 
9:   end for
10: end function

```

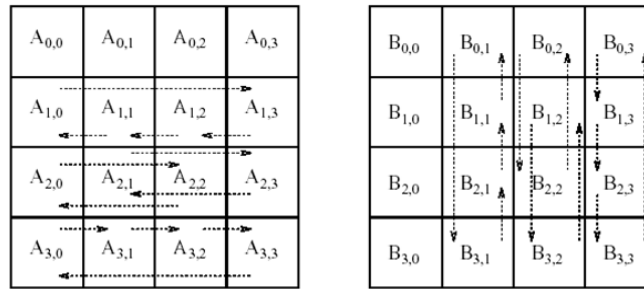


Figura 1: Exemplu de rulare a Algoritmului 1 – alinierea inițială

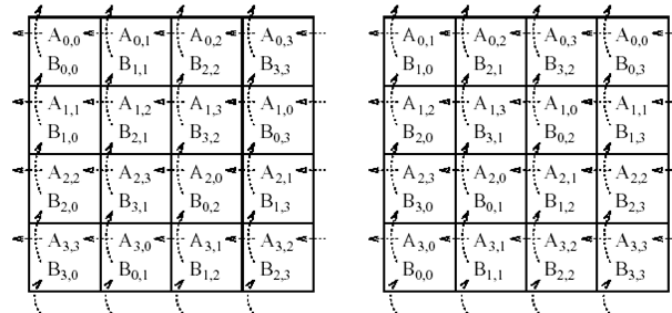


Figura 2: Exemplu de rulare a Algoritmului 1 – A și B după alinierea inițială (stânga), pozițiile blocurilor după prima deplasare (dreapta)

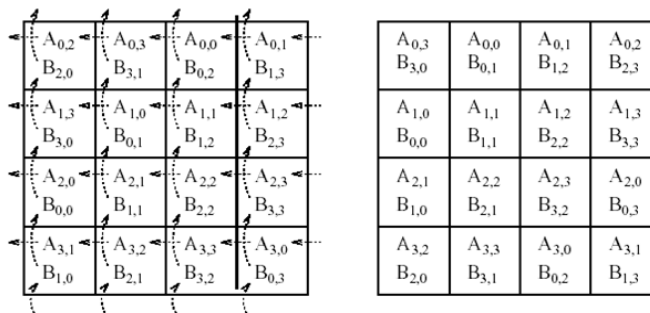


Figura 3: Exemplu de rulare a Algoritmului 1 – pozițiile blocurilor după a II-a deplasare (stânga), pozițiile blocurilor după a III-a deplasare (dreapta)

2 Aplicații

Implementați utilizând MPI Algoritmul lui Cannon pentru înmulțirea a două matrice. Se va utiliza o topologie de tip plasă de procese.