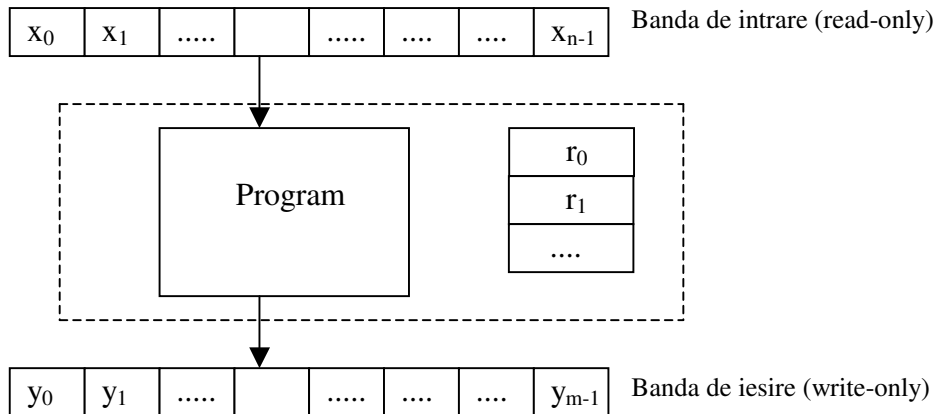


Laborator 1

Modele teoretice de calculatoare paralele

Modelul RAM

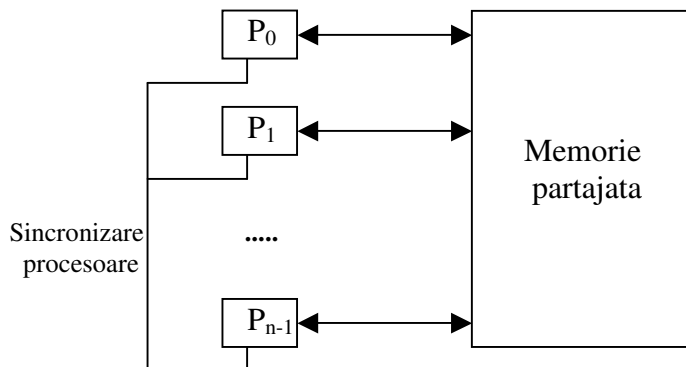
Modelul RAM (Random Access Machine) de calculator secvential (Sheperdson si Sturgis):



Memoria consta dintr-un numar nelimitat de registre $r_0, r_1, r_2, r_3, \dots$ pentru numere intregi. Dupa fiecare citire / scriere banda de intrare / iesire avanseaza o pozitie. Timpul de executie al programului depinde de numarul de date de intrare n .

Modelul PRAM

Modelul PRAM (Parallel RAM) pentru un calculator paralel cu cost suplimentar nul de sincronizare si acces la memoria (Fortune si Wylie):



Optiunile de acces concurrent al procesoarelor la memoria partajata sunt:

-ER (Exclusiv Read): intr-un ciclu cel mult un procesor poate sa citeasca dintr-o locatie de memorie;

-EW (Exclusiv Write): intr-un ciclu cel mult un procesor poate sa scrie intr-o locatie de memorie;

-CR (Concurrent Read): mai multe procesoare pot sa citeasca din aceeași locatie de memorie in același ciclu;

-CW (Concurrent Write): mai multe procesoare pot sa scrie in aceeași locatie de memorie in același ciclu.

Rezulta urmatoarele variante ale modelului PRAM:

-EREW-PRAM: este cel mai restrictiv;

-CREW-PRAM: conflictele de scriere sunt evitate prin excludere mutuala;

-ERCW-PRAM;

-CRCW-PRAM: este cel mai puternic model.

Conflictele de scriere concurenta se rezolva printr-unul din urmatoarele moduri:

-COMMON-PRAM: toate operatiile de scriere memoreaza aceeași valoare la locatia accesata simultan;

-ARBITRARY-PRAM: se memoreaza una din valori (oricare), iar celelalte sunt ignorate;

-MINIMUM-PRAM: se memoreaza valoarea scrisa de procesorul cu indexul cel mai mic;

-PRIORITY-PRAM: se memoreaza o valoare obtinuta prin aplicarea unei functii asociative (exemplu: insumare) tuturor valorilor cu care se acceseaza locatia de memorie.

Algoritm secvential pentru inmultire de matrici

```
for (i=0; i<n; i++)
  for (j=0; j<n; j++) {
    c[i][j]=0;
    for (k=0; k<n; k++)
      c[i][j]+=a[i][k]*b[k][j];
  }
```

Algoritmul necesita timpul de executie $T_s = O(n^3)$.

Algoritm CRCW-PRAM cu n^3 procesoare pentru inmultire de matrici

Conflictele de scriere se rezolva cu modelul PRIORITY-PRAM folosind operatorul de insumare. Matricile a , b , c sunt in memoria partajata. Cele n^3 procesoare sunt plasate intr-un tablou 3D. Algoritmul se executa in doua etape:

Etapa 1: fiecare procesor $P_{i,j,k}$ calculeaza in paralel produsul $a[i][k]*b[k][j]$.

Etapa 2: se face scrierea concurenta a procesoarelor $P_{i,j,0}$; $P_{i,j,1}$; $P_{i,j,2}$; ...; $P_{i,j,n-1}$ in locatia $c[i][j]$ a termenilor $a[i][0]*b[0][j]$, $a[i][1]*b[1][j]$, $a[i][2]*b[2][j]$, ... , $a[i][n-1]*b[n-1][j]$, care astfel se insumeaza.

Algoritmul paralel este:

```
forall (0 <= i < n)
  forall (0 <= j < n)
    forall (0 <= k < n) {
      c[i][j] = 0.0;
      c[i][j] = a[i][k] * b[k][j];
    }
}
```

Timpul de executie paralela $T_p = O(1)$. Rezulta accelerarea $S = O(n^3)$ si eficienta $E = 1$.

Algoritm CREW-PRAM cu n^3 procesoare pentru inmultire de matrici

Procesoarele sunt organizate intr-un tablou 3D. Este necesar un tablou suplimentar v cu $n*n*n$ elemente in memoria partajata in locul matricii c . Algoritmul se executa in doua etape:

Etape 1: fiecare procesor $P_{i,j,k}$ calculeaza in paralel produsul $a[i][k]*b[k][j]$ si inscrie rezultatul in $v[i][j][k]$.

Etape 2: reducerea prin insumarea a n elemente ale fiecarui vector din tabloul 2D $v[i][j]$ (in total n^2 vectori), folosind n procesoare pentru fiecare vector: $P_{i,j,0}; P_{i,j,1}; P_{i,j,2}; \dots; P_{i,j,n-1}$, obtinand rezultatul in $v[i][j][0]$.

Algoritmul este:

```
forall (0 <= i < n)
  forall (0 <= j < n)
    forall (0 <= k < n) {
      v[i][j][k] = a[i][k] * b[k][j];
      for (m=0; m < log2n; m++)
        if (k % 2^{m+1} == 0)
          v[i][j][k] += v[i][j][k+2^m];
    }
}
```

Fiecare procesor executa un pas de calcul si $\log_2 n$ pasi pentru reducerea paralela, deci timpul de executie este:

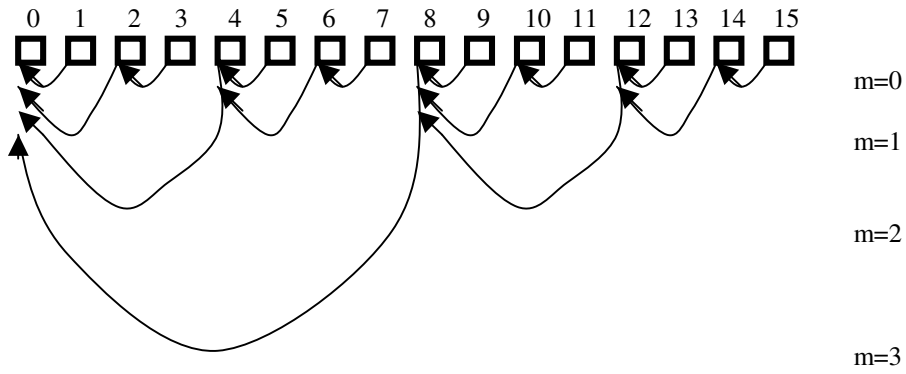
$$T_p = 1 + \log_2 n = O(\log_2 n)$$

fiind cuprins intre $O(n^3)$ in cadrul algoritmului secvential si $O(1)$ in algoritmul CRCW. Se calculeaza accelerarea si eficienta:

$$S = O(n^3 / \log_2 n)$$

$$E = O(1 / \log_2 n)$$

Operatia de reducere paralela se poate efectua astfel (exemplu pentru 16 procesoare si 4 pasi):



Tema

Sa se scrie un program care sa simuleze executia algoritmilor CRCW-PRAM si CREW-PRAM pentru produsul de matrici. Sa se analizeze un algoritm ERCW-PRAM pentru produsul de matrici.