

## PARALELNI SISTEMI

1. Projektovati četvorostepeni protočni sistem za izračunavanje elemenata niza C na osnovu izraza:

$$c_i = \frac{si \cdot ri(h_i - e_i)(h_i + e_i)}{f_i^5 - \frac{a_i b_i}{d_i} - c_{i-1} f_i}, i=1,2,\dots, n; c_0=0. \text{ Svaki PE u sistemu može da obavlja osnovne aritmetičke operacije i}$$

naredbu NOP. Na raspolaganju su procesni elementi sa i bez lokalne memorije.

a) Odrediti format mikronaredbe opisanog sistema

b) Napisati mikroprogram, kao i sadržaj lokalnih memorija koji mu odgovara, za izračunavanje prva dva elementa niza C. Obratiti pažnju na optimalnost rešenja.

2. Koristeći CUDA tehnologiju, sastaviti program koji na osnovu matrice  $A_{N \times N}$  kreira vektor  $B_N$ , gde je element  $B[i]$  jednak  $P \cdot Q$ , gde je  $P$  broj pozitivnih elemenata  $i$ -te vrste matrice  $A$ , a  $Q$  broj negativnih elemenata  $i$ -te vrste matrice  $A$ . Zatim, sve elemente manje od nule  $i$ -te vrste matrice  $A$ , zameniti vrednošću  $B[i]$ . Maksimalno redukovati broj pristupa globalnoj memoriji. Obratiti pažnju na efikasnost paralelizacije. Omogućiti rad programa za matrice proizvoljne veličine.

3. a) Napisati MPI program koji realizuje množenje matrice  $A_{n \times n}$  i vektora  $b_n$ , čime se dobija rezultujući vektor  $c_n$ . Matrica  $A$  i vektor  $b$  se inicijalizuju u master procesu. Broj procesa je  $p$  i uređeni su kao matrica  $q \times q$  ( $q^2=p$ ). Matrica  $A$  je podeljena u blokove i master proces distribuira odgovarajuće blokove matrice  $A$  po procesima kao što je prikazano na slici 1. za  $n=8$  i  $p=16$ . Vektor  $b$  je distribuiran po procesima tako da proces  $P_i$  dobija elemente sa indeksima  $i \% q, i \% q + q, i \% q + 2q, \dots, i \% q + n - q$ . Predvideti da se slanje vrednosti bloka matrice  $A$  svakom procesu obavlja odjednom. Svaki proces (uključujući i master) obavlja odgovarajuća izračunavanja i učestvuje u generisanju rezultata koji se prikazuje u procesu koji sadrži minimum svih vrednosti u matrici  $A$ . Predvideti da se slanje blokova matrice  $A$  svakom procesu obavlja sa po jednom naredbom `MPI_Send` kojom se šalje samo 1 izvedeni tip podatka. Slanje blokova vektora  $b$  i generisanje rezultata implementirati korišćenjem grupnih operacija i funkcija za kreiranje novih komunikatora.

$a_{00}$	$a_{04}$	$a_{01}$	$a_{05}$	$a_{02}$	$a_{06}$	$a_{03}$	$a_{07}$
P0		P1		P2		P3	
$a_{10}$	$a_{14}$	$a_{11}$	$a_{15}$	$a_{12}$	$a_{16}$	$a_{13}$	$a_{17}$
P4		P5		P6		P7	
P8		P9		P10		P11	
$a_{60}$	$a_{64}$	$a_{61}$	$a_{65}$	$a_{62}$	$a_{66}$	$a_{63}$	$a_{67}$
P12		P13		P14		P15	
$a_{70}$	$a_{74}$	$a_{71}$	$a_{75}$	$a_{72}$	$a_{76}$	$a_{73}$	$a_{77}$

Slika1.

b) Slanje blokova matrice  $A$  svakom procesu implementirati korišćenjem grupnih operacija i funkcija za kreiranje novih komunikatora

4. a) Napisati deo MPI koda koji formira komunikator Cartesian topologije, koji omogućava da sadržaj promenljive  $x$ , u svakom procesu, pre i nakon poziva funkcije:

```
MPI_Sendrecv_replace(&x, 1, MPI_INT, dest, 0, source, 0, cartcomm, &st);
```

izgleda kao na prikazanom primeru komunikatora 3x4.

Vrednosti  $x$  pre:

0	1	2	3
1	2	3	4
2	3	4	5

Vrednosti  $x$  posle:

0	1	2	3
4	1	2	3
4	5	2	3

b) Napisati OpenMP kod koji sadrži sledeću petlju:

```
z=s;  
for ( i = 0; i < N; i++ )  
    for ( j = 0; j < N; j++ )  
    {  
        A[j] += B[z];  
        z+=k;  
    }
```

i proučiti da li moguće izvršiti njenu paralelizaciju. Ako nije izvršiti njenu transformaciju tako da paralelizacija bude moguća. Vrednosti za promenljive  $s$  i  $k$ , kao i za elemente nizova  $A$  i  $B$  su inicijalizovane pre petlje. Nakon petlje treba prikazati vrednosti za elemente niza  $A$  i za promenljivu  $z$ , generisanim u okviru petlje. Testiranjem sekvencijalnog i paralelnog rešenja za proizvoljno  $N$  i proizvoljan broj niti, pokazati korektnost paralelizovanog koda.