

PARALELNI SISTEMI

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

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

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, napisati program koji određuje rezultujuću matricu $B_{N/2 \times N/2}$ na osnovu matrice $A_{N \times N}$, na sledeći način:

$$B(i,j) = \text{avg}(A(i,j), A(i,j+2), A(i+2,j), A(i+2,j+2))$$

Ilustracija rada programa za $N = 5$ data je na slici. Maksimalno redukovati broj pristupa globalnoj memoriji. Obratiti pažnju na efikasnost paralelizacije.

Primer:

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} & A_{0,4} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,0} & A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{bmatrix} = \begin{bmatrix} \text{avg}(A_{0,0}, A_{0,2}, A_{2,0}, A_{2,2}) & \dots & \dots \\ \text{avg}(A_{1,0}, A_{1,2}, A_{3,0}, A_{3,2}) & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

3. Napisati MPI program koji realizuje množenje matrice $A_{m \times m}$ i vektora b_m , čime se dobija rezultujući vektor c_m . 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 za $m=8$ i $p=16$. Vektor b je distribuiran po procesima tako da proces P_i dobija elemente sa indeksima $i \% q, i \% q + q, i \% q + 2q, \dots, i \% q + m - q$. 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 maksimum 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.

a00 a04 a08 a12 a16 a20 a24 a28 a32 a36 a40 a44 P0	a01 a05 a09 a13 a17 a21 a25 a29 a33 a37 a41 a45 P1	a02 a06 a10 a14 a18 a22 a26 a30 a34 a38 a42 a46 P2	a03 a07 a11 a15 a19 a23 a27 a31 a35 a39 a43 a47 P3
P4	P5	P6	P7
P8	P9	P10	P11
a30 a34 a38 a42 a46 P12	a31 a35 a39 a43 a47 P13	a32 a36 a40 a44 P14	a33 a37 a41 a45 P15
a70 a74 a78 P16	a71 a75 a79 P17	a72 a76 a80 P18	a73 a77 a81 P19

15

4. a) Napisati MPI kod 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 4x4.

Vrednosti x pre:

0	1	2	3
1	2	3	4
2	3	4	5
3	4	5	6

Vrednosti x posle:

0	1	2	3
2	3	4	1
4	5	2	3
6	3	4	5

b) Napisati OpenMP kod koji sadrži sledeću petlju:

```
v = start ;
sum = 0;
for ( int i = 0; i < N ; i++)
{
    sum = sum + f ( v ) ;
    v = v + step ;
}
```

i proučiti da li moguće izvršiti njenu paralelizaciju. Ako nije izvršiti njenu transformaciju tako da paralelizacija bude moguća. Vrednosti za promenljive $start$ i $step$, kao i kod koji realizuje funkciju f su poznati pre petlje. Nakon petlje treba prikazati vrednosti za promenljive v i sum , generisanim u okviru petlje. Testiranjem sekvencijalnog i paralelnog rešenja za proizvoljno N i proizvoljan broj niti, pokazati korektnost paralelizovanog koda.