

## PARALELNI SISTEMI

1. Koristeći CUDA tehnologiju, napisati C/C++ program koji računa zbir dva vektora proizvoljne dužine, i pronalazi srednju vrednost i standardnu devijaciju rezultujućeg vektora. Pretpostaviti veličinu bloka od 256 niti i broj blokova ne veći od 256. Maksimalno redukovati broj pristupa globalnoj memoriji. Obratiti pažnju na efikasnost paralelizacije.

Formula za izračunavanje standardne devijacije:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

gde je:

N - broj elemenata u skupu,  $\mu$  - aritmetička sredina skupa,  $x_i$  - i-ti član skupa ( $i=1,2,\dots,N$ )

2. a) Napisati MPI program koji realizuje množenje matrice  $A_{k \times k}$  i vektora  $b_k$ , čime se dobija rezultujući vektor  $c_k$ . 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  $k=8$  i  $p=16$ . Vektor b je distribuiran po procesima tako da proces  $P_i$  dobija elemente sa indeksima  $(i \% q) * s, (i \% q) * s + 1, \dots, (i \% q) * s + s - 1$  ( $s=k/q$ ). 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 <sub>00</sub> a <sub>01</sub>	a <sub>02</sub> a <sub>03</sub>	a <sub>04</sub> a <sub>05</sub>	a <sub>06</sub> a <sub>07</sub>
P0	P1	P2	P3
a <sub>40</sub> a <sub>41</sub>	a <sub>42</sub> a <sub>43</sub>	a <sub>44</sub> a <sub>45</sub>	a <sub>46</sub> a <sub>47</sub>
P4	P5	P6	P7
P8	P9	P10	P11
a <sub>30</sub> a <sub>31</sub>	a <sub>32</sub> a <sub>33</sub>	a <sub>34</sub> a <sub>35</sub>	a <sub>36</sub> a <sub>37</sub>
P12	P13	P14	P15
a <sub>70</sub> a <sub>71</sub>	a <sub>72</sub> a <sub>73</sub>	a <sub>74</sub> a <sub>75</sub>	a <sub>76</sub> a <sub>77</sub>

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

3. Napisati MPI program koji vrši paralelni upis i čitanje binarne datoteke, prema sledećim zahtevima:

1. Svaki proces upisuje po 105 proizvoljnih celih brojeva u datoteku *file1.dat*. Upis se vrši upotrebom pojedinačnih pokazivača, dok redosled podataka u fajlu ide od podataka poslednjeg do podataka prvog procesa.
2. Ponovo otvoriti datoteku. Svaki proces vrši čitanje upravo upisanih podataka upotrebom funkcija sa eksplicitnim pomerajem. Proveriti ispravnost pročitanih podataka.
3. Upravo pročitane podatke upisati u novu datoteku, na način prikazan na slici (za slučaj od 3 procesa):



U poslednjem zahtevu posebno obratiti pažnju na efikasnost paralelizacije upisa.

4. Napisati OpenMP kod koji sadrži sledeću petlju:

```
for (i = N - 1; i > 1; i--)  
{  
    x = res[i] + add[i];  
    sum[i] = sum[i - 1] + x;  
}
```

i proučiti da li je moguće izvršiti njenu paralelizaciju. Ako nije, transformisati petlju tako da paralelizacija bude moguća. Nakon petlje treba prikazati vrednosti za promenljive x i sum, generisanim u okviru petlje. Testiranjem sekvencijalnog i paralelnog rešenja za proizvoljno N i proizvoljan broj niti, pokazati korektnost paralelizovanog koda.