

Paralelno programiranje - završni ispit

Napomene uz programske zadatke: tekst programa (pseudokod) pisati čitko i strukturirano (uvlačenje). Obvezno kratko opisati idejno rješenje programa.

- [] 1. (6) Podatkovna struktura u 3 dimenzije sadrži $N \times N \times N$ elemenata. Za izračun nove vrijednosti svakog elementa potrebne su vrijednosti 6 elemenata koji ga neposredno okružuju (iznad, ispod, lijevo, desno, ispred i iza). Trošak računanja jednog elementa iznosi t_c , a parametri komunikacije su t_s i t_w . Izrazite *trajanje izvođenja* jedne iteracije na P procesora te *učinkovitost* i *izoučinkovitost* ako su podaci po procesorima podijeljeni po sve 3 dimenzije (tako da svaki procesor ima jednak broj elemenata).
- [] 2. (4) Trajanje nekog programa dano je u dvije komponente: dio programa koji se mora izvoditi slijedno ima trajanje $10 \cdot N$, a dio programa koji se može idealno paralelizirati ima trajanje N^2 (idealna paralelizacija uz P procesora skraćuje trajanje P puta). Ako nam je na raspolaganju budžet od 500€, a paralelno računalo se plaća 1€ po satu po procesoru, koliko je najkraće moguće trajanje programa za $N=10$?
- [] 3. (5) Za autonomnu letjelicu dostupan je niz podataka u kojemu svaki element predstavlja trenutnu poziciju letjelice u jednoj prostornoj osi (trenutnu udaljenost od polazišne točke). Vremenski razmak između dva mjerenja (dvije pozicije u nizu) je jedna sekunda, pozicija je zapisana u metrima, a polazišna točka je na poziciji 0. Potrebno je odrediti najveću akceleraciju (u m/s^2) kojoj je letjelica bila izložena za vrijeme kretanja (max_acc) i u kojoj sekundi kretanja (max_t). Primjerice, ako je niz pozicija [1 3 3 -1 0 2 5 4], tada je niz akceleracija [1 1 2 4 5 1 4]. Napišite algoritam za EREW PRAM koji rješava ovaj problem za niz podataka duljine n uz najviše n procesora.
- [] 4. (5) U jednom trenutku svaki proces MPI programa ima lokalnu varijablu v . Korištenjem MPI funkcija *Send* i *Recv* (skraćena sintaksa) napisati odsječak programa *logaritamske vremenske složenosti* (po pitanju broja poslanih poruka) koji će za N procesa provjeriti čini li vektor vrijednosti lokalnih varijabli v svakog procesa permutacijski vektor (permutacija skupa $\{1, 2, \dots, n\}$). Primjerice, (3, 1, 2) je permutacijski vektor dok (2, 4, 3), (1, 1, 2) i (4, 2, 1) nisu. Varijabla v poprima samo prirodne vrijednosti, a rezultat provjere treba biti poznat na svakom procesu.
- [] 5. (3) Napišite naredbu koja treba doći na mjesto crte kako bi kako bi rezultat izvođenja ove *kernel* funkcije bio zbroj svih elemenata polja `d_vect` (*točnost sintakse se neće uzimati u obzir pri bodovanju*). Varijabla `size` predstavlja veličinu vektora, a skup dretvi definiran je u jednoj dimenziji. Kratko argumentirajte svoj odgovor.
- ```

__global__ void funkcija(int* d_vect, size_t size, int* result){
 size_t tid = blockIdx.x * blockDim.x + threadIdx.x;
 while(tid < size){

 }
}

```
- [ ] 6. (3) Usporedite osnovne razlike u arhitekturi CPU-a i GPU-a u kontekstu skupa instrukcija, procesne moći (broja i brzine jezgara) te memorijske strukture.
- [ ] 7. (8) Paralelni algoritam izvodi operaciju `+_prescan` niza  $A[]$ , gdje svaki od  $P$  zadataka s indeksom  $ID$  posjeduje jedan element niza ( $A[ID]$ ). Trošak izvođenja jedne operacije zbrajanja za dva elementa je  $t_c$ , a parametri komunikacije za pojedine elemente su  $t_s$  i  $t_w$ . Koliko traje izvođenje operacije `+_prescan` ako se koristi:
- (2) komunikacijska struktura lanca (svaki zadatak prima međurezultat od prethodnog zadatka, računa novi međurezultat i šalje ga sljedbeniku),
  - (2) komunikacijska struktura binarnog stabla (broj zadataka je potencija broja 2)? Nacrtajte stablo.
  - (4) Korištenjem MPI funkcija *Send*(*<podatak>*, *<odredište>*) i *Recv*(*<podatak>*, *<izvorište>*), napišite odsječak programa za oba načina izvođenja operacije `+_prescan`.



[ ] 8. (4) Parovi vrijednosti zapisani su u obliku niza znakova u formatu *ključ:vrijednost* te međusobno odvojeni znakom ';'. Znakovi su pohranjeni u nizu  $A[]$  duljine  $n$ . Npr. niz znakova  $A[] = [\text{jabuka:crvena}; \text{kruska:zelena}; \text{limun:zuta}]$  duljine 38 znakova sadrži 3 para. Napišite algoritam za EREW PRAM računalo uz najviše  $n$  procesora koji će ispisati koliko ukupno ima takvih parova vrijednosti.

[ ] 9. (4) Navedite moguće uzroke neslaganja (neprikladnosti) jednostavnog modela trajanja paralelnog programa u usporedbi sa stvarnom izvedbom programa.

[ ] 10. (1) Navedite barem četiri načina ujednačavanja opterećenja u fazi pridruživanja: \_\_\_\_\_

(1) Ako se ujednačavanje opterećenja provodi samo jednom (na početku izvođenja), tada to nazivamo \_\_\_\_\_

(1) Povećavanje zrnatosti može se postići \_\_\_\_\_  
ili \_\_\_\_\_

(1) Načini kompozicije paralelnih modula su \_\_\_\_\_

(1) Navedite barem četiri parametra operatora migracije u raspodijeljenom evolucijskom algoritmu: \_\_\_\_\_

(1) Ako je ubrzanje linearno, učinkovitost je (kojeg iznosa?) \_\_\_\_\_

(1) Ako je učinkovitost 75%, a broj procesora je 12, koliko iznosi ubrzanje? \_\_\_\_\_

(1) Ukupan broj poruka koji se razmijeni u provedbi komunikacijske strukture binarnog stabla (npr. algoritam reduciranja) dubine  $d$  iznosi \_\_\_\_\_