

1. UVOD

1. Nepoželjno svojstvo paralelnog programa je „**podatkovna ovisnost**“.
2. Nedostaci MIMD modela paralelnog računala su:
 - A. odvojene su instrukcije za skalarne i vektorske operande
 - B. sva grananja se primjenjuju na sve procesore
 - C. **općenito teže za programirati**
 - D. procesori su sinkronizirani po instrukcijama
3. Prednosti MIMD modela paralelnog računala su:
 - A. **moguće je izvoditi različite nizove instrukcija**
 - B. **jednom instrukcijom paralelno obrađujemo više podataka**
 - C. sva grananja se primjenjuju na sve procesore
 - D. procesori su sinkronizirani po instrukcijama
4. Nedostatci SIMD modela: **Grananja unutar petlji moraju se primijeniti na sve procesore.**
5. Koje tvrdnje su istinite:
 - A. **u modelu raspodijeljene memorije procesori imaju vlastiti spremnik**
 - B. u modelu zajedničke memorije komunikacija je moguća jedino razmjenom poruka
 - C. u modelu raspodijeljene memorije više procesora koristi isti spremnik
 - D. **u modelu raspodijeljene memorije programer je odgovoran za dijeljenje podataka**
6. Poželjna svojstva paralelnih programa su **istodobnost, skalabilnost, lokalnost i modularnost.**
7. Amdahlov zakon definira iznos najvećeg mogućeg **ubrzanja** u ovisnosti o **dijelu programa koji se može paralelizirati.**
8. Zrnatost zadataka se može definirati kao **omjer između količine računanja (lokalnog rada) i količine komunikacije (nelokalnog rada).**
9. U modelu raspodijeljene memorije, procesori mogu komunicirati jedino **razmjenom poruka.**
10. Amdahlov zakon povezuje **slijedni i paralelni dio programa, broj procesora i ubrzanje.**

2. MPI

11. Jednom procesoru može biti dodijeljeno **više** MPI procesa.
12. Povratak iz neblokirajuće MPI funkcije znači: **da se može ponovno pristupiti toj memorijskoj lokaciji, ali ne i da je funkcija uspješno izvršena – to se mora naknadno provjeriti.**
13. Povratak iz blokirajuće MPI funkcije znači **da je ista završila u smislu da se sigurno može pristupati memorijskim lokacijama korištenim u komunikaciji.**
14. MPI mehanizam dijeljenja komunikatora omogućava izvedbu **paralelne** kompozicije modula u paralelnom programu.
15. MPI mehanizam modula u paralelnim programima omogućava izvedbu **slijedne i paralelne kompozicije modula**

3. PRAM

16. Složenost provedbe postupka scan niza duljine n elemenata na PRAM računalu uz p procesora gdje je $p < (n/2)$ iznosi **$O(n/p + \log p)$**
17. Prilikom izvođenja optimalno postupka +_reduciranja niza duljine n na PRAM računalu, ukupan broj operacija zbrajanja na svim procesorima iznosi **$n-1$.**

18. Prilikom prilagodbe PRAM algoritma za APRAM računalo uz (p/B) procesora, gdje jedan APRAM procesor izvodi instrukcije za B PRAM procesora, jedna EREW PRAM instrukcija izvodi se u **$5B+2d-2$ koraka**. Napomena: **$(2B+2d+1)$ je za p procesora, vidjeti predavanja 26. str.)**
19. Algoritam scan se odnosi na bilo koju **binarnu asocijativnu** operaciju.
20. Navedite i ukratko opišite barem 3 primjene prescan algoritma:
- Najveći element u nizu** - na zadanom nizu izvedemo postupak reduciranja uz operator $\max()$ koji prima dva argumenta i vraća većega: \max reduciranje
 - Provjera uređenosti niza** - pridijelimo procesor svakom elementu niza i svaki procesor provjerava je li njegov element manji ili jednak sljedećemu i rezultat zapisuje kao 1 ili 0. Na dobivenom vektoru izvedemo andprescan i provjerimo vrijednost zadnjeg elementa (zapravo je dovoljno i andreduciranje)
 - Alokacija procesora** - npr. imamo zadan vektor zahtjeva za memorijom za tri procesa (element vektora govori koliko memorije traži određeni proces): $[4\ 1\ 3]$ Kako odrediti početne adrese memorijskih segmenata? Rješenje se dobiva +_prescan postupkom: $[0\ 4\ 5]$
21. Inačica PRAM računala koja omogućuju istovremeno čitanje iste memorijske lokacije je **CRCW**.
22. Prilikom istodobnog čitanja iste memorijske lokacije u CRCW PRAM računalu, svaki procesor će pročitati **istu/jednaku** vrijednost
23. Vremenska složenost provedbe postupka scan niza duljine n elemenata na PRAM računalu uz p procesora gdje je $p < n/2$, iznosi **$O(n/p + \log p)$** .

4. APRAM

24. Na APRAM računalu, uz trajanje globalnog pristupa 4 vremenske jedinice, 2 uzastopna globalna pristupa trajat će **$5 = 4 + 2 - 1$ ($d+k-1$)**.
25. Navedite i opišite dodatne parametre koji definiraju svojstva APRAM računala (u odnosu na PRAM računalo).
- d** odnos vremena globalnog i lokalnog pristupa memoriji
 - $B=B(p)$** vrijeme potrebno za sinkronizaciju svih p procesora
26. Na APRAM računalu, unutar istog asinkronog odsječka, samo **jedan** procesor smije pristupiti **istoj globalnoj** memorijskoj lokaciji.
27. Optimalna složenost algoritma reduciranja niza duljine n na APRAM računalu uz n procesora iznosi **$O(b \log n)$ (pomoću d -arnog stabla umjesto binarnog)**.
28. Vrste instrukcija na APRAM računalu su: **globalno čitanje, globalno pisanje, lokalna operacija i sinkronizacija(barrier)**.
29. Na APRAM računalu, uz trajanje globalnog pristupa 3 vremenske jedinice, 4 uzastopna globalna pristupa trajat će **6 ($3+4-1$ tj. $(d+k-1)$)**.

5. Postupak oblikovanja paralelnih algoritama

30. Pridruživanje se provodi ukoliko je broj **zadataka(nakon aglomeracije)** veći od broja **procesora**.
31. Faza pridruživanja se provodi ako je broj **zadataka** veći od broja **procesora**.
32. Uvišestručavanje računanja je tehnika kojom se **povećava** ukupna količina računanja kako bi se **smanjila** količina komunikacije
33. Navedite sve podjele komunikacije u paralelnim algoritmima: **Lokalna i globalna komunikacija; strukturirana/nestrukturirana; statička/dinamička; sinkrona/asinkrona**.

34. Povećanje zrnatosti može se postići tehnikama: **povećanje zadataka, uvišestručavanje računanja.**
35. Pojava zagušenja voditelja u modelu voditelj-radnik moguća je uz **prevelik** broj radnika.
36. Ukupan broj poruka koje se razmijene (pošalju) u provedbi komunikacijske strukture binarnog stabla za 2D procesora iznosi **$2^d - 1$**
37. Ukupan broj poruka koje se razmijene (pošalju) u provedbi komunikacijske strukture hiperkocke za 2^d procesa iznosi **$n \cdot \log(n)$** .
38. Sitnozrnata podjela posla podrazumijeva malu količinu komunikacije u odnosu na veću količinu računanja. **NETOČNO.** (sitnozrnata znači puno zadataka od kojih je svaki malen, tj. malo računanja, puno komunikacije)
39. Prilikom pridruživanja zadataka procesorima, zadatke koji se izvode neovisno poželjno je pridružiti istom procesoru. **NETOČNO** (to je osnova za paralelizaciju - neovisni zadaci na različitim procesorima. Iz skripte: zadaci koji se izvode neovisno/istodobno dodjeljuju se različitim procesorima | | zadaci koji često komuniciraju dodjeljuju se istom procesoru)
40. Navedi i objasni tehnike ujednačavanja opterećenja.
 - A. **Statičko – zadaci se raspoređuju samo na početku rada.**
 - B. **Dinamičko – ako se mijenjaju uvjeti izvođenja, ujednačavanje se pokreće nekoliko puta tijekom rada paralelnog programa**

6. Kvantitativna analiza paralelnog programa

41. Ukoliko se ubrzanje paralelnog programa mjeri u odnosu na najbolji slijedni program, radi se o **apsolutnom** ubrzanju
42. Ukoliko se ubrzanje paralelnog programa mjeri u odnosu na isti program pokrenut na jednom procesoru, radi se o **relativnom** ubrzanju.
43. Ako su ostali parametri isti i povećavamo broj procesora, **učinkovitost monotono pada, a ubrzanje raste do neke točke pa nakon toga pada.**
44. Ako je učinkovitost 25%, a ubrzanje je 4, koliki je broj procesora? **16?** ($S = P \cdot E$ – $S = \text{ubrzanje}$, $E = \text{učinkovitost}$ pa onda $P = S/E$ tj. $4/0.25 = 16$)
45. Izoučinkovitost opisuje kako se mora promijeniti br. procesora u ovisnosti o količini posla (računanja) kako bi **učinkovitost** ostala nepromijenjena
46. Ako je ubrzanje linearno, učinkovitost je **jednaka 1.**
47. Ako je ubrzanje superlinearno, učinkovitost je **veća od 1.**
48. Ako je ubrzanje sublinearno, učinkovitost je **manja od 1.**
49. Ubrzanje veće od linearnog naziva se **superlinearno.**
50. Kako bi ubrzanje bilo veće od 1, trajanje komunikacije i čekanja mora biti kraće od računanja na pojedinom procesoru. **TOČNO**
51. Trajanje računanja (T_r) paralelnog programa može ovisiti o raspodjeli zadataka po procesorima. **TOČNO**
52. Funkcija izoučinkovitosti opisuje kako se mora promijeniti **količina posla** u ovisnosti o promjeni broja procesora kako bi **učinkovitost** ostala nepromijenjena.
53. Izraz koji opisuje trajanje slanja jedne poruke duljine L riječi u jednostavnom modelu komunikacije je: **$T_{\text{msg}} = t_s + t_w \cdot L$ (t_s - postavljanje poruke, t_w - prijenos jedne riječi).**
54. Navedite moguće nedostatke jednostavnog modela ocjene performansi u opisu stvarnog ponašanja paralelnog programa? **Jednostavni model ne uzima u obzir ograničenja bandwidth-a komunikacijskog kanala, nejednako opterećene, uvišestručeno računanje, nesklad algoritma i programskog alata.**
55. Navedite moguće uzroke neslaganja (nepotpunosti) jednostavnog modela trajanja paralelnog programa u usporedbi sa stvarnom izvedbom programa: **Nejednako opterećenje,**

uvišestručeno računanje, nesklad algoritma i programskog alata, ograničen kapacitet komunikacije.

56. Navedite i moguće uzroke “anomalije ubrzanja”: **Upotreba priručne memorije (cache-a) i anomalija pretraživanja.**
57. Ukupno trajanje računanja (Tr) paralelnog programa može ovisiti o raspodjeli zadataka po procesorima. **TOČNO** (u raznorodnom sustavu, trajanje računanja ovisi i o raspodjeli zadataka po procesorima)
58. Trajanje izvođenja paralelnog programa ne ovisi o promatranom procesoru iz skupa svih procesora koji izvode paralelni program. **TOČNO.**
59. Trajanje izvođenja paralelnog programa ovisi o promatranom procesoru iz skupa svih procesora. **NETOČNO**
60. Trajanje izvođenja paralelnog programa je po definiciji neovisno o promatranom procesoru. **TOČNO**
61. Uz povećanje količine računanja i nepromijenjene ostale elemente, učinkovitost opada. **NETOČNO.**
62. Uz smanjenje količine računanja i nepromijenjene ostale elemente računanja, učinkovitost programa raste. **TOČNO**

7. Paralelno programiranje grafičkih procesora

63. Navedi 3 vrste memorije GPU i objasni razlike između njih.

A. Globalna memorija – radna memorija jednog uređaja

- Najveća, najsporija
- Sve dretve mogu pristupiti sadržaju
- Koristi se za komunikaciju s uređajem domaćinom (host) – jedina memorija kojoj je moguće pristupiti izvan uređaja
- 8-16 GB

B. Lokalna memorija – radna memorija jednog multiprocesora

- Puno manja, ali brža od globalne memorije
- Mogu joj pristupiti samo dretve koje se izvode na dotičnom (istom) multiprocesoru
- Tipično reda veličine 48-64 Kb

C. Privatna memorija – radna memorija jedne dretve

- Najmanja veličina, najbrži pristup

8. Razvoj modularnih paralelnih programa

64. Kompozicija modula u paralelnim programima može biti: **Slijedna, paralelna i zajednička kompozicija.**

9. Paralelni evolucijski algoritmi

65. Navedi dva načina paralelizacije evolucijskih algoritama i navedi parametre koji se koriste kod njih.

A. Standardni pristup

- Paralelizirati genetske operatore i izračunati vrijednosti funkcije cilja paralelno

B. Dekompozicijski pristup

- Podijeliti populaciju na manje dijelove – subpopulacije i obavljati cijeli genetski algoritam nad subpopulacijama