

Teorija s prethodnih MI

1. Zrnatost zadataka se može definirati kao **omjer između količine računanja (lokalnog rada) i količine komunikacije (nelokalnog rada)**.
2. U modelu raspodijeljene memorije, procesori mogu komunicirati jedino **razmjenom poruka**.
3. Složenost provedbe postupka scan niza duljine n elemenata na PRAM računalu uz p procesora gdje je $p \leq (n/2)$ iznosi **$O(n/p + \log p)$** .
4. Prilikom izvođenja optimalno postupka $+$ _reduciranja niza duljine n na PRAM računalu, ukupan broj operacija zbrajanja na svim procesorima iznosi **$n-1$** .
5. Na APRAM računalu, uz trajanje globalnog pristupa 4 vremenske jedinice, 2 uzastopna globalna pristupa trajat će **$5 = 4 + 2^{-1} (d+k - 1)$** .
6. Prilikom prilagodbe PRAM algoritma za APRAM računalu 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.)
7. Optimalna složenost algoritma reduciranja niza duljine n na APRAM računalu uz n procesora iznosi **$O(B \log n)$** ili **$O(B \log n / \log B) \rightarrow O(\log_B n)$** (pomoću B -arnog umjesto binarnog stabla).
8. Algoritam scan se odnosi na bilo koju **binarnu asocijativnu** operaciju.
9. Poželjna svojstva paralelnih programa su **istodobnost, skalabilnost, lokalnost i modularnost**.
10. Povratak iz neblokirajuće MPI funkcije ne znači **da je funkcija uspješno izvršena (nego da se može ponovo pristupiti memorijskoj lokaciji)**.
11. Vrste instrukcija na APRAM računalu su **globalno čitanje, globalno pisanje, lokalna operacija i sinkronizacija**.
12. Amdahlov zakon definira iznos najvećeg mogućeg **ubrzanja** u ovisnosti o **dijelu programa koji se može paralelizirati**.

13. Navedite i ukratko opišite barem 3 primjene (pre)scan algoritma:

- a. **Najveći element u nizu** - na zadanom nizu izvedemo postupak reduciranja uz operator $\max()$ koji prima dva argumenta i vraća većega:
max-reduciranje
- b. **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 and-prescan i provjerimo vrijednost zadnjeg elementa (zapravo je dovoljno i and-reduciranje)
- c. **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]

14. Navedite i opišite dodatne parametre koji definiraju svojstva APRAM računala (u odnosu na PRAM računalo).

- i. **d** - odnos vremena globalnog i lokalnog pristupa memoriji
- ii. **$B=B(p)$** - vrijeme potrebno za sinkronizaciju svih p procesora