

Fakultet elektrotehnike i računarstva
Zavod za primjenjeno računarstvo

Paralelno programiranje

2. laboratorijska vježba

Mato Manović 0036484071

Zagreb, 26.5.2018.

Četiri faze razvoja:

1. Podjela

Podjela se jednostavno može napraviti tako da provjera podstabala tj. provjera sljedećih stanja se dekomponira u podzadatke. Svaki zadatak će pomoću neke definirane funkcije dobrote odrediti koliko je taj potez dobar i vratiti rezultat. Na kraju će se ti rezultati samo usporediti i odrediti potez koji daje najveću vrijednost funkcije dobrote te će stoga i biti odabran kao sljedeći potez. Zadaci su međusobno neovisni tj. nije potrebna nikakva podjela podataka između njih.

2. Komunikacija

Faza izračunavanja u svakom zadatku je neovisna o drugima, stoga nije potrebno imati nikakvu komunikaciju. Jedina potrebna komunikacija je ona koja je potrebna za njihovu sinkronizaciju tj. dodjelu zadataka te prikupljanje rezultata nakon obrade. Preciznije, lokalna komunikacija ne postoji, a globalna komunikacija se koristi samo za razmjenu zadataka i rezultata istih.

3. Aglomeracija

Zrnatost zadataka se jednostavno podešava pomoću dva konfigurabilna parametra, a to su TASK_DEPTH_INIT i TASK_DEPTH. Konkretno prvi parametar predstavlja dubinu do koje ćemo obavljati podjelu zadataka. Pošto u igri Connect4 imamo mogućih 7 poteza to znači da će broj zadataka biti potencija broja 7. Na primjer za TASK_DEPTH_INIT=3 dobivamo $7^3=343$ zadataka. S drugim parametrom jednostavno definiramo složenost svakog zadatka tj. do koje će dubine svaki zadatak pretraživati prostor stanja. Ukupno dubina pretraživanja je onda suma tih dvaju parametara.

4. Pridruživanje

Način raspoređivanja zadataka je klasični *voditelj/radnik* model. Dakle, u ovakvom modelu jedan proces predstavlja voditelja tj. obavlja stvaranje zadataka, raspoređivanje zadataka radnicima i obavljanje komunikacije s njima te na kraju objedinjavanje rezultata svih izvedenih zadataka. Broj zadataka je konfigurabilan tj. možemo stvoriti više zadataka nego procesa što nam pomaže pri ujednačavanju opterećenja pošto svaki zadatak se ne mora izvršavati isto vrijeme, zbog *branch and boundinga* kojeg provodimo tj. neka stanja brzo mogu odvesti u stanje pobjede ili poraza.

Empirijska evaluacija ubrzanja i učinkovitosti

Rezultati u tablici dobiveni su mjerenjem na računalu na procesoru AMD FX-8150 s osam jezgri. Vremena predstavljaju koliko je vremena potrebno računalu da odgovori na protivnički potez. Izmjerena vremena su na početku igre s dva odigrana poteza 2 i 3. Mjerenja su uprosječena te su potom izračunata učinkovitost i ubrzanje, te korigirana učinkovitost gdje je za bazno vrijeme na jednom procesoru uzet rezultat izvršavanja na dva procesora, pošto su ta vremena ekvivalentna jer u slučaju dva procesora i više proces voditelj ne obavlja nikakav posao osim komunikacije. Vremena su izražena u milisekundama, inicijalna dubina podjele na zadatke je 2 tj. ukupno je 49 zadataka te svaki zadatak pretražuje prostor stanja do dubine 7 što je ukupna dubina pretraživanja 9. Rezultati su navedeni u tablici te grafički prikazani.

Broj procesora	1. mjerjenje	2. mjerjenje	prosjeak	Učinkovitost E	Ubrzanje S	Korigirani E*
1	27722	27053	27388	1	1	-
2	28543	27921	28232	0.485	0.970	1
3	19446	18847	19147	0.477	1.431	0.737
4	13248	13567	13408	0.510	2.040	0.701
5	9906	9761	9833	0.557	2.785	0.717
6	7892	7710	7801	0.585	3.510	0.723
7	6389	6411	6400	0.611	4.277	0.735
8	5635	5570	5603	0.611	4.888	0.719

