

2. laboratorijska vježba

Paralelno programiranje

1. Opis razvoja algoritma

Kako se zadatak - implementacija računalnog algoritma za igranje igre **4 u nizu** - rješava pretraživanjem stabla igre i biranjem najboljeg poteza, a pretraživanje pojedine grane sasvim neovisno od ostalih grana, algoritam je moguće, uz prethodnu jednostavnu funkcijsku dekompoziciju, trivijalno paralelizirati.

1.1 Podjela

Algoritam pretraživanja stabla igre funkcijski se, kada ga se nastoji paralelizirati, može rastaviti na tri faze: **stvaranje podzadataka**, **obrada pojedinih podzadataka** i **objedinjavanje rezultata** svih obrada, i pri tome se **obrada pojedinih podzadataka** može vršiti neovisno, tj. paralelno. **Broj podzadataka**, kao i njihova **složenost**, su **varijabilni**, ovisno o dubini stabla igre na kojoj se vrši podjela (tako da sve grane koje započinju na toj dubini postaju zasebni podzadaci, svi jednako složeni), te dubini pretraživanja pojedine grane. Ostale dvije faze izvršavaju se serijski, na jednom za to zaduženom računalu, i nad njima se ne vrši podjela.

1.2 Komunikacija

Kako su podzadaci koji se paralelno obrađuju **trivijalno paralelizirani**, između njih nije potrebna nikakva komunikacija, već je jedina prisutna komunikacija ona između **glavnog procesa**, koji izvršava prvu i treću fazu, i **radnika** koji obrađuju pojedine podzadatke. Ona se primarno sastoji od **upita radnika** za zadatkom za obradu te njihova **slanja odgovora** glavnom procesu, uz još poneku **upravljačku poruku** poput naloga za buđenje, spavanje te zaustavljanje rada, koje glavni proces šalje radnicima.

1.3 Aglomeracija

Razina aglomeracije – zrnatosti zadataka – definirana je preko konfigurabilnih parametara unutar koda programa (`#define BRANCH_DEPTH 2 => 72 zadataka`, `#define TASK_DEPTH 6 => dubina pretraživanja = složenost pojedinog zadatka`), a isto se tako može prilagoditi i za postavljanje prilikom poziva programa. Time je omogućeno da se, smanjenjem parametra `BRANCH_DEPTH`, smanji i broj zadataka, a time i količina komunikacije za istu ukupnu razinu pretraživanja.

1.4 Pridruživanje

Pojedini zadaci, kojih ovisno o konfiguraciji programa može biti **različit broj**, pridružuju se procesima koji će ih obrađivati prema modelu *manager/worker*, gdje je **jedan proces** zadužen za **stvaranje i raspodjelu zadataka**, a ostali procesi su **radnici** koji obrađuju pojedinačne dodijeljene im zadatke. U

ovaj je pristup ugrađeno i **ujednačavanje opterećenja**, budući da je broj zadataka u pravilu znatno veći (iako je to sve konfigurabilno) od broja radnika, pa oni radnici koji u nekom trenutku imaju više dostupnih resursa, stižu obaviti veći broj zadataka od ostalih, pa su neiskorišteni računalni resursi minimalni.

2. Empirijsko mjerenje ubrzanja i učinkovitosti

U tablici su prikazani rezultati dobiveni mjerenjem na dva računala (C13 i C14) iz labosa D337. Vremenske vrijednosti predstavljaju trajanje računanja pojedinog poteza računala, za iste poteze korisnika (1 → 2 → 3 → 4).

Osim izravnih rezultata mjerenja te srednje izmjerene vrijednosti, u tablici se nalaze i ubrzanje **S**, učinkovitost **E**, te korigirana učinkovitost **E***, koja u obzir uzima činjenicu da uvođenjem drugog procesora (i procesa), prvi prestaje i sam obrađivati zadatke te preuzima isključivo zadaću glavnog procesa, što se očituje i u gotovo identičnim vremenima za izvršavanje na 1 i 2 procesora.

Broj procesora	Trajanje koraka					S	E	E*
	1.	2.	3.	4.	AVG			
1	12.24	12.34	11.71	9.83	11,53	1,00	1,00	---
2	12.27	12.37	11.74	9.86	11,56	1,00	0,50	1,00
3	6.27	6.28	5.96	4.98	5,87	1,96	0,65	0,98
4	4.28	4.27	4.03	3.40	3,99	2,89	0,72	0,97
5	3.29	3.28	3.07	2.61	3,06	3,77	0,75	0.94
6	2.61	2.60	2.49	2.08	2,44	4,73	0,79	0.95
7	2.34	2.31	2.18	1.85	2,17	5,31	0,76	0.89
8	1.96	1.94	1.83	1.58	1,83	6,30	0,79	0,90

Rezultati trajanja prikazani su i grafički, a tako i ubrzanje, učinkovitost, te korigirana učinkovitost.

Uočljivo je kako se na *1. grafu* koji prikazuje **vremena trajanja**, ona ponašaju **inverzno** u odnosu na **broj procesora**, te da se za iste korake performanse ponašaju jednako.

Na *2. grafu* koji prikazuje **ubrzanje** vidljivo je da ono, uz prethodno objašnjenu iznimku pri povećanju s jednog na dva procesora, **raste linearno**.

Na *3. grafu* koji prikazuje **učinkovitosti**, vidljiv je pad pri uvođenju drugog procesora, čiji je razlog prethodno objašnjen, i on dovodi do **prepolavljanja** početne učinkovitosti. Kako se broj procesora povećava, tako se taj efekt uvođenja glavnog procesa sve više smanjuje, i pri broju od 6-8 procesora njegov utjecaj postaje nezamjetan.

Na tom je grafu također prikazana i **korigirana učinkovitost**, koja kao baznu vrijednost uzima **trajanje izvršavanje** programa na **dva procesora**, gdje jedan vrši posao glavnog programa, a drugi radnika, i

tamo je zamjetno da se učinkovitost od početne vrijednosti 1 **polagano smanjuje** zbog povećanja utjecaja komunikacije spram računanja.



