

Paralelno programiranje – 2. domaća zadaća

Zadatak

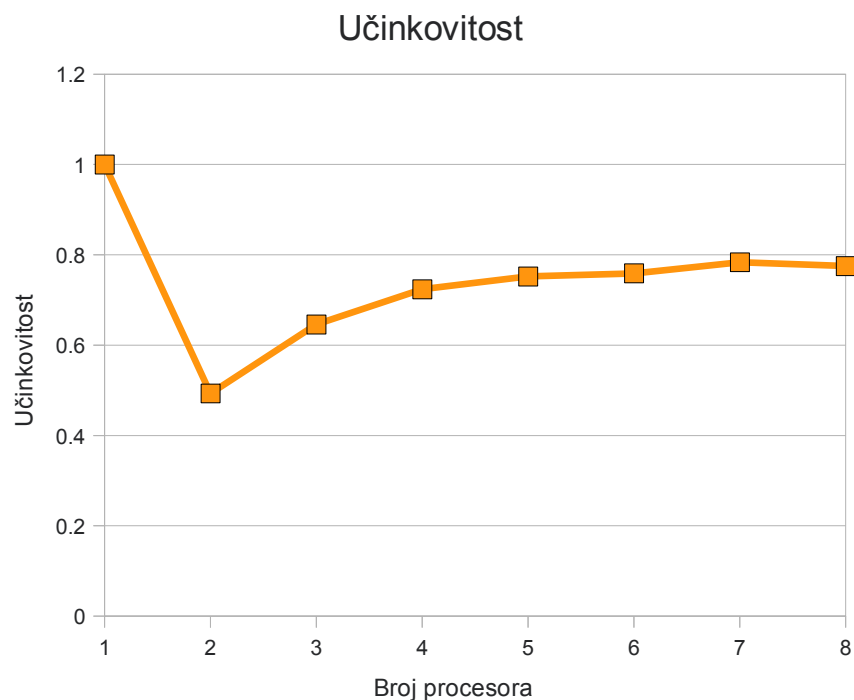
U sklopu ove domaće zadaće bilo je potrebno empirijski utvrditi ubrzanje i učinkovitost predloženog algoritma za djelomično pretraživanje prostora stanja u igri “4 u nizu”. To je učinjeno mjerenjem trajanja programa (odnosno trajanja računanja jednog poteza) na $P=1,\dots,8$ identičnih procesora.

Mjerenja

# proc.	M1	M2	Avg	Učinkovitost	Ubrzanje
1	30,41	30,36	30,38	1,00	1,00
2	30,79	30,83	30,81	0,49	0,99
3	15,58	15,78	15,68	0,65	1,94
4	10,48	10,50	10,49	0,72	2,90
5	8,13	8,03	8,08	0,75	3,76
6	6,91	6,44	6,67	0,76	4,55
7	5,55	5,53	5,54	0,78	5,49
8	5,00	4,80	4,90	0,78	6,20

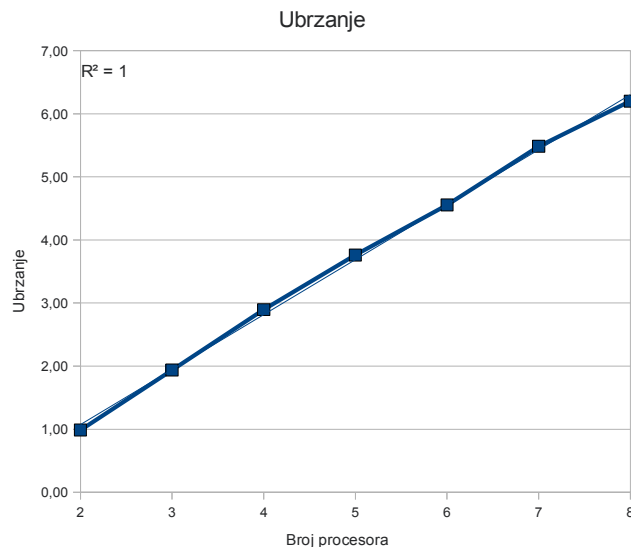
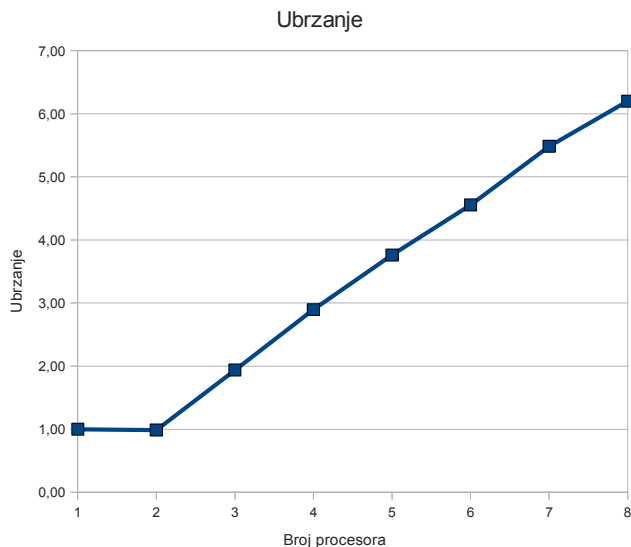
Za svaku vrijednost parametra p učinjena su 2 mjerenja kako bi se isključila mogućnost da je jedno izvođenje slučajno bilo brže ili sporije od ostalih.

Učinkovitost



Vidljivo je da je učinkovitost za slijedni algoritam jednaka 1, što je i očekivano s obzirom na definiciju učinkovitosti. Za 2 procesora učinkovitost drastično opada, na vrijednost oko 0.5, što je također očekivano s obzirom da u slučaju sa 2 i više procesora jedan procesor služi isključivo za komunikaciju (“master-slave” princip). Također, iz grafa se vidi kako povećanjem broja procesora učinkovitost raste, što je logično jer se smanjuje utjecaj master procesora.

Ubrzanje



Na desnom grafu koji prikazuje ubrzanje paralelnog algoritma u slučaju sa 2 ili više procesora vidimo da ono linearno raste s obzirom na broj procesora. To je svakako željeni rezultat. Ukoliko bi na grafu bili vidljivi skokovi ili ravni dijelovi, to bi značilo da algoritam ne iskorištava veći broj procesora na ispravan način. Primjerice, na lijevom grafu između točaka 1 i 2 postoji ravna crta zbog činjenice da u slučaju sa 2 procesora cijeli jedan procesor posvećujemo sinkronizaciji i komunikaciji pa ne dolazi do nikakvog ubrzanja, već se nešto malo performansi gubi zbog potrebe za sinkronizacijom.

Zaključak

Iz grafova učinkovitosti i ubrzanja možemo zaključiti da je predloženi algoritam dobro skalabilan i učinkovit. Vidljivo je da učinkovitost raste povećanjem broja procesora i da se malena stagnacija primjećuje tek kod $P=8$. Ubrzanje je linearno, bez naglih skokova, ravnih dijelova ili padova, što je i željeno ponašanje.

Stoga je moguće zaključiti da je navedeni algoritam dobro prilagođen za paralelni rad.