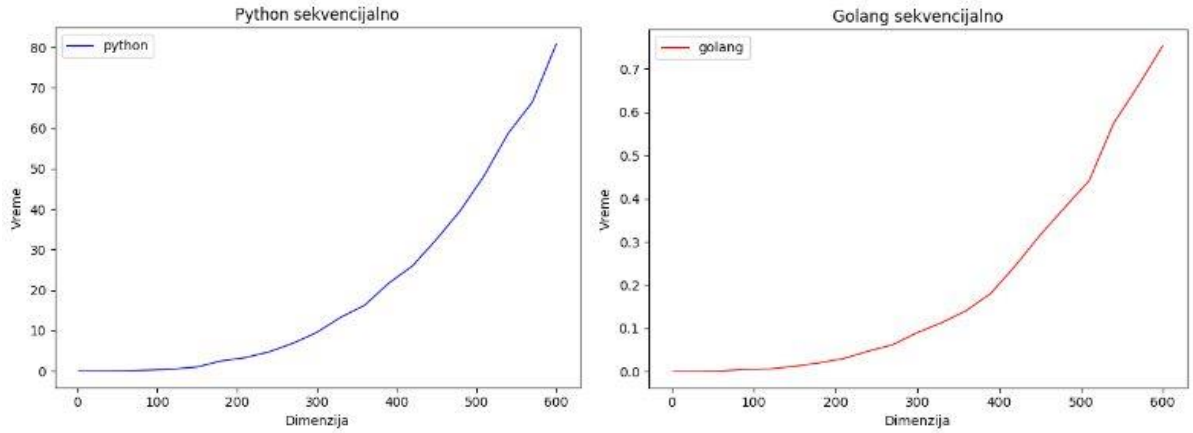
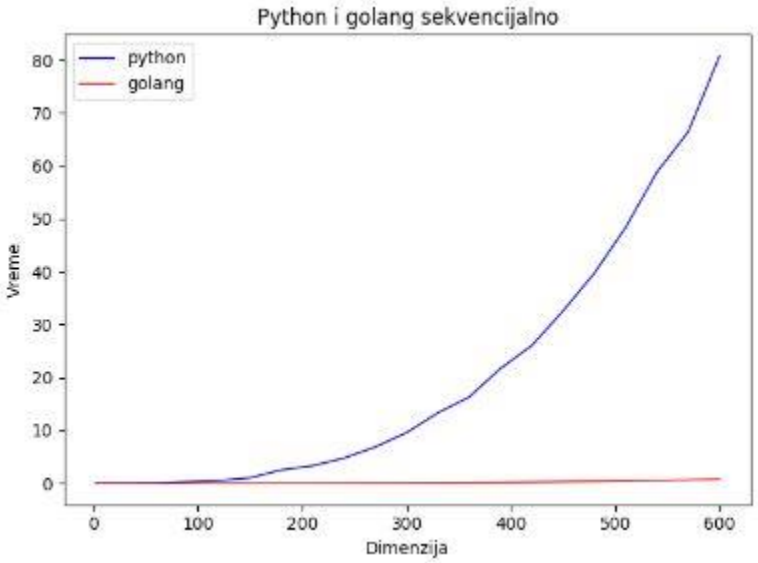
Kanonov algoritam za množenje matrica

Poređenje sekvencijalnih rešenja





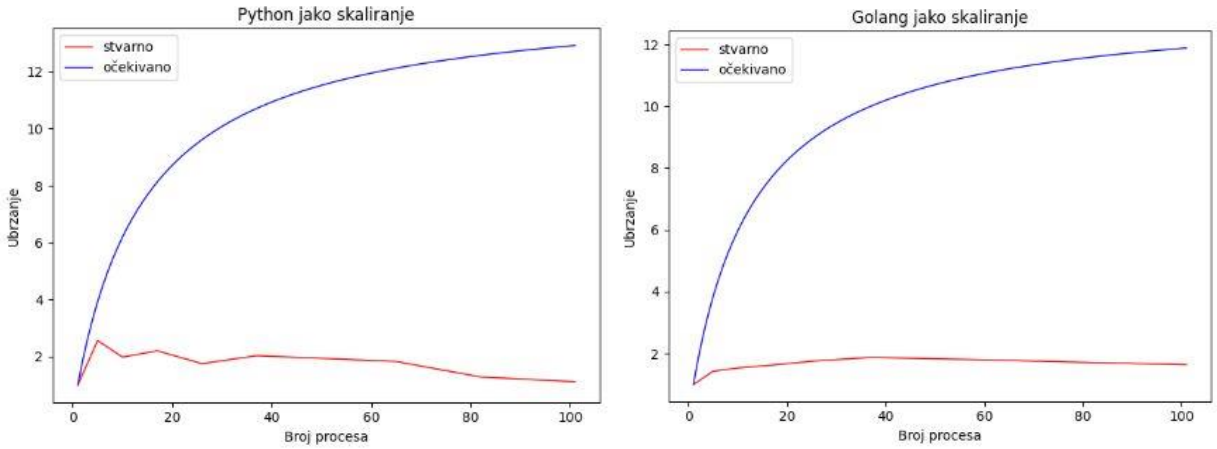
Na graficima iznad je prikazano vreme koje je potrebno (y-osa) da se pomnože dve matrice uz pomoć Kanonovog algoritma (sekvencijalno rešenje), dimenzija nxn, pri čemu je n prikazano na x-osi. Sa slika se može uočiti da go jezik mnogo brže dolazi do rešenja nego python. Kod malih dimenzija razlika nije toliko primetna, dok kod većih dimenzija jeste, na primer u python-u za množenje matrica dimenzija 600x600 je potrebno 80 sekundi, dok je go jeziku potrebno manje od sekunde.

Poređenje paralelnih rešenja

Eksperimenti su rađeni na računaru sa procesorom Intel(R) Core(TM) i5-5200U CPU, radni takt 2.20GHz, L1 cache: 128 KB, L2 cache: 512KB, L3 cache: 3.0 MB, broj fizičkih jezgara: 2 jezgra, broj logičkih jezgara: 4, memorija 8.0GB DDR3.

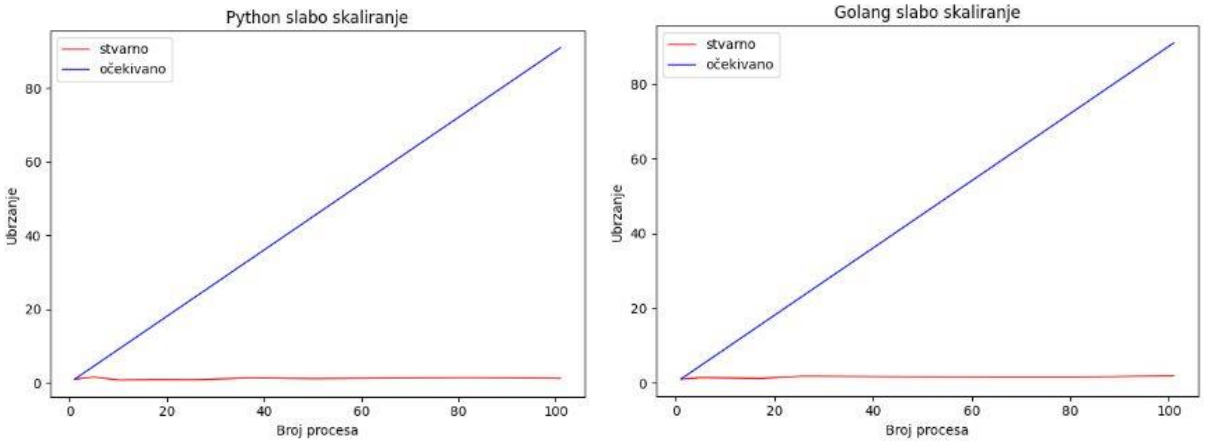
Procenat sekvencijalnog dela koda se razlikuje kod jakog i slabog skaliranja, razlog tome je što se procenat sekvencijalnog dela koda razlikuje u zavisnosti od dimenzije matrice. Određivanje sekvencijalnog procenta kod jakog skaliranja je bilo jednostavnije jer je korišćena jedna dimenzija matrice, odnosno 360, a dobijeni procenat je 7,5. Kod slabog skaliranja procenat je 10, izabran je na osnovu najčešće dobijanih procenata za dimenzije koje su se koristile.

Jako skaliranje



Na graficima iznad je prikazano ubrzanje (y-osa) koje se ostvaruje korišćenjem paralelne implementacije množenja dve matrice uz pomoć Kanonovog algoritma, dimenzija nxn, pri čemu n iznosi 360. Paralelno rešenje se sastoji od jednog glavnog procesora i još p procesora. Vrednosti p na kojima je beleženo vreme i računato ubrzanje su: 4, 9, 16, 25, 36, 64, 81, 100. Kao i u prethodnom slučaju go je imao bolje vremenske rezultate. Sa grafika se primeti da se u python-u najveće ubrzanje postiže kada je p jednako 4, kao i da ima veće ubrzanje kada je p deljivo sa 4. Kod go jezika najveće ubrzanje je kad je p 36, do tog momenta ubrzanje raste, a nakon toga kreće da pada.

Slabo skaliranje



Na graficima iznad je prikazano ubrzanje (y-osa) koje se ostvaruje korišćenjem paralelne implementacije množenja dve matrice uz pomoć Kanonovog algoritma, dimenzija nxn. Paralelno rešenje se sastoji od jednog glavnog procesora i još p procesora, na x-osi su date vrednosti p. Vrednosti p na kojima je beleženo vreme su: 4, 9, 16, 25, 36, 49, 64, 81, 100. Dimenzije matrice su zavisile od p, odnosno svakom procesoru p je bio dodeljen deo matrice dimenzija 50x50, kako bi se postiglo da u svakom procesu postoji jednak broj posla. Do dimenzija matrice se dolazi tako što se 50 pomnoži sa korenom od p, na primer ako je p jednako 4, n je bilo 100, a ako je p bilo 25, n je bilo 250. Kao i u prethodna dva slučaja go jezik mnogo brže dolazi do rešenja nego python. Sa grafika se vidi da kod oba jezika sa porastom broja procesora raste i ubrzanje, dok je očekivano ubrzanje u rasponu od 0 do 91, stvarno ubrzanje je u rasponu od 0 do 2.

Implementacija paralelnih verzija

Implementacija paralelne verzije u programskom jeziku python odrađena je uz pomoć biblioteke mpi4py. Glavni procesor dodeljuje ostalim p procesorima delove matrica kojim oni upravljaju, pre dodele glavni procesor izvrši inicijalno pomeranje. P procesori nakon toga rade neophodna množenja i pomeranja, a kada završe vraćaju rezultat koji su dobili za njima dodeljen deo glavnom procesoru, koji sve spaja u jednu matricu. Implementaciju paralelne verzije u go programskom jeziku odrađena je uz oslonac na goroutine. Nema većih razlika u odnosu na python, sam postupak je identičan. Go(lang) za razliku od python-a čiji su svi procesi istovremeno pokrenuti, svoje procese (niti) poziva iz glavnog procesa. U python-u procesi komuniciraju uz pomoć send-recv mehanizma, gde proces mora da zna kom procesu šalje podatke prilikom send, takođe mora da zna i od kog procesa prima podatke prilikom recv. Go(lang) koristi kanale za komunikaciju, jedan proces upisuje podatke u jedan kanal, dok drugi čita iz istog tog kanala, pre pokretanja procesa određuju mu se kanali u koje upisuje i iz kojih čita podatke.