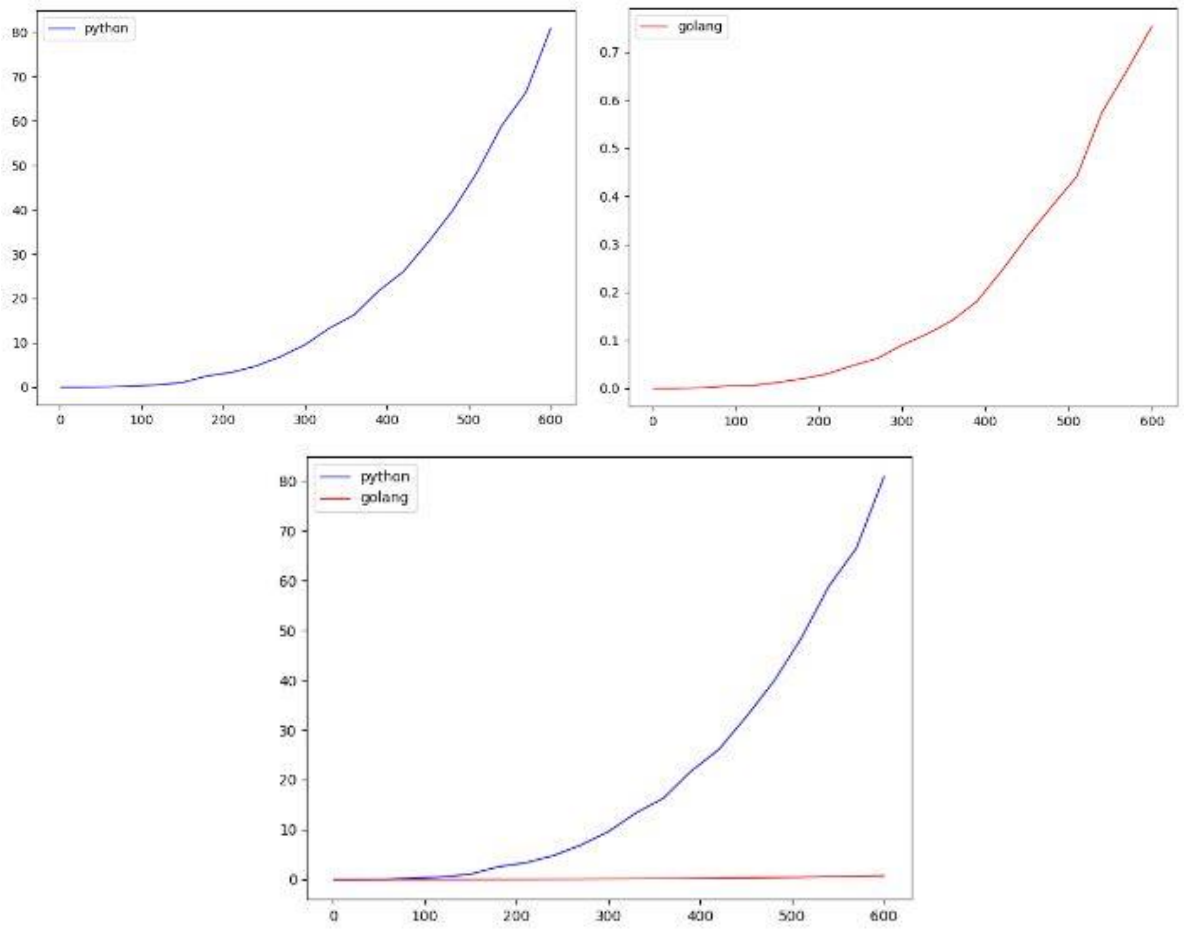
Kanonov algoritam za množenje matrica

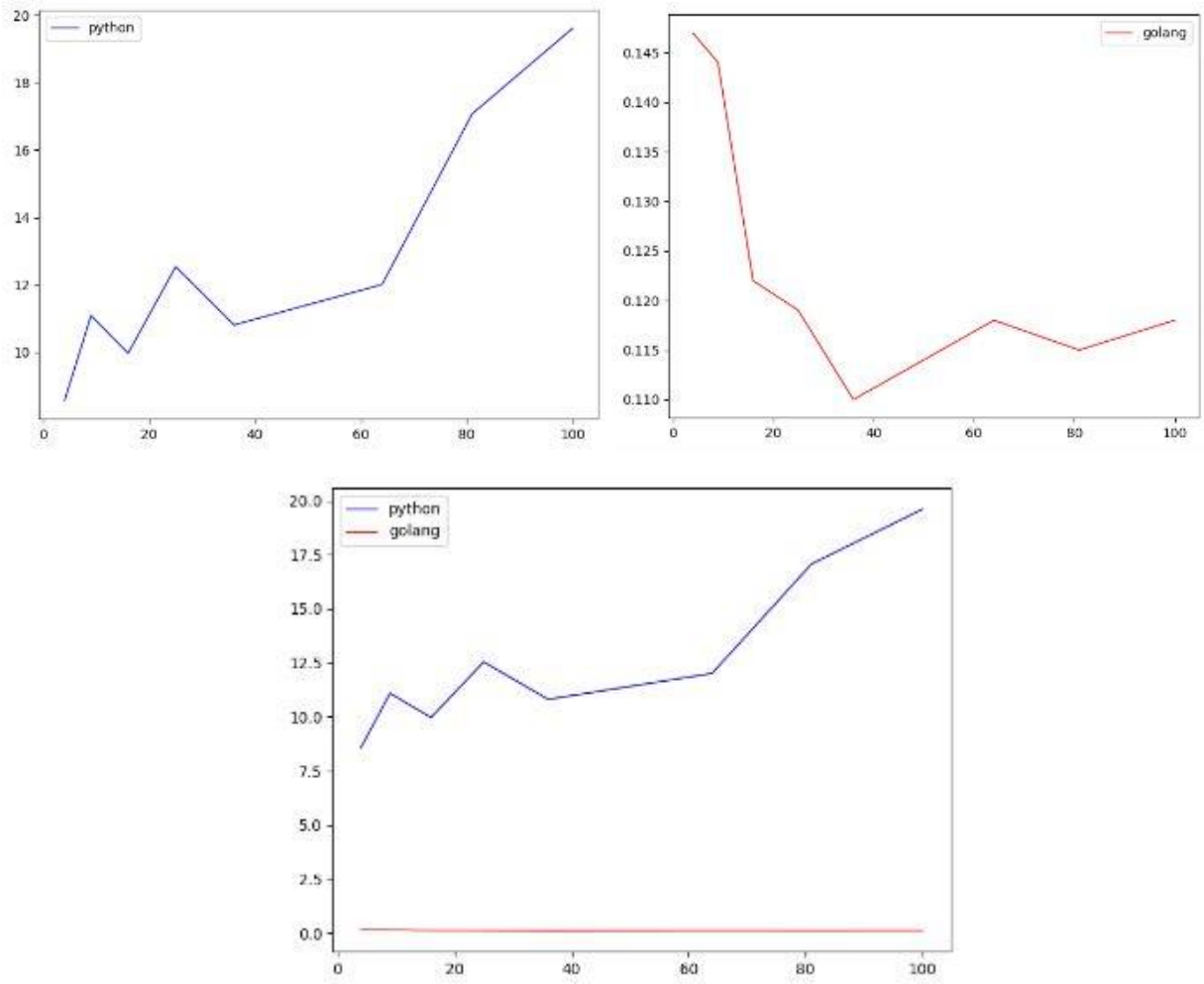
Poređenje sekvencijalnih rešenja



Na slikama 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.

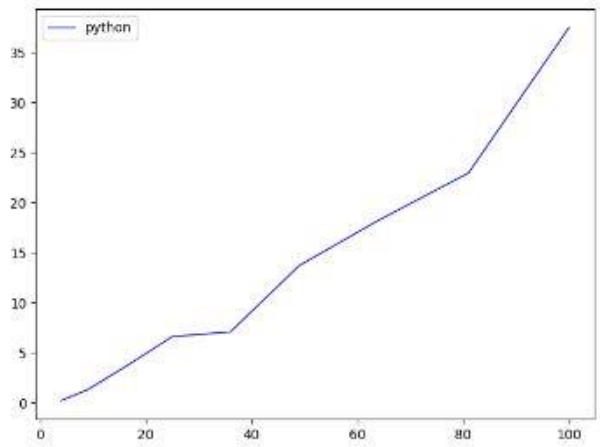
Poređenje paralelnih rešenja

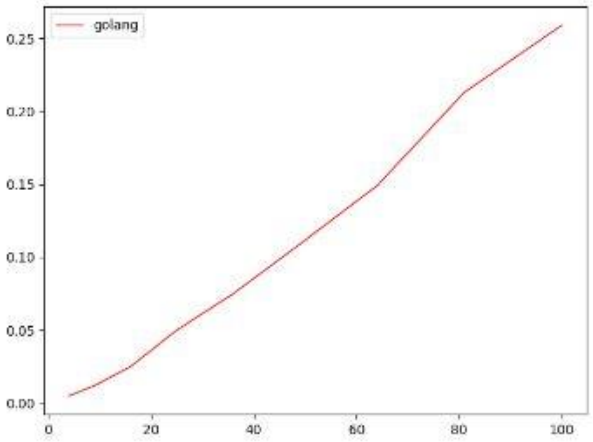
Jako skaliranje

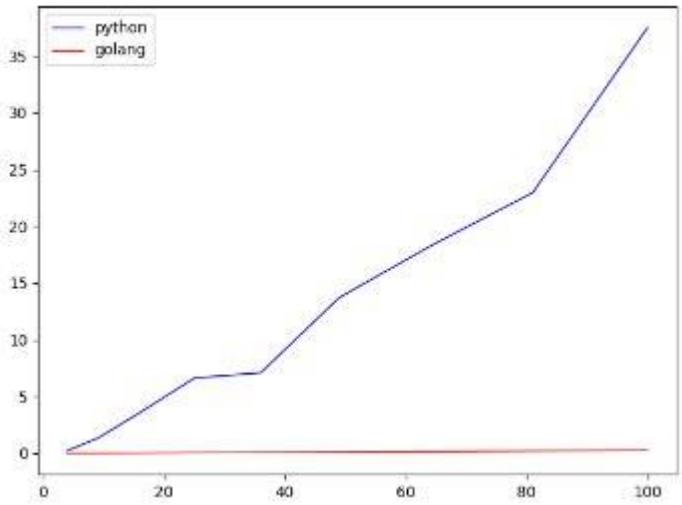


Na slikama iznad je prikazano vreme koje je potrebno (y-osa) da se pomnože dve matrice uz pomoć Kanonovog algoritma, dimenzija nxn, pri čemu n iznosi 360. Paralelno rešenje se sastoji od jednog glavnog procesora i još p procesora, na x-osi su date vrednosti p. Vrednosti p na kojima je beleženo vreme su: 4, 9, 16, 25, 36, 64, 81, 100. Kao i u prethodnom slučaju može se primetiti da go jezik mnogo brže dolazi do rešenja nego python. Ono što se može uočiti jeste da kod oba jezika ne postoji pravilo po kom se može odrediti optimalna veličina p, odnosno ne postoji garancija da će veće ili manje p davati bolje rezultate.

Slabo skaliranje







Na slikama iznad je prikazano vreme koje je potrebno (y-osa) da se pomnože 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. 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 može se primetiti da go jezik mnogo brže dolazi do rešenja nego python. Sa slika se vidi da kod oba jezika sa porastom broja procesora, raste i vreme koje je neophodno za dobijanje rešenja, što je i logično, jer su veće dimenzije same matrice, a i postoji više komunikacija između procesa.

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.