

UNIVERZITET U BEOGRADU - ELEKTROTEHNIČKI FAKULTET
MULTIPROCESORSKI SISTEMI (13S114MUPS, 13E114MUPS)



Domaći zadatak 2 - MPI

Izveštaj o urađenom domaćem zadatku

Predmetni saradnici:

doc. dr Marko Mišić

dipl. ing. Pavle Divović

Studenti:

Iva Potkonjak 2019/0301

Lazar Premović 2019/0091

Beograd, decembar 2022.

Sadržaj

1	PROBLEM 1 - PRIME	2
1.1	Tekst problema	2
1.2	Delovi koje treba paralelizovati	2
1.2.1	Diskusija	2
1.2.2	Način paralelizacije	2
1.3	Rezultati	3
1.3.1	Logovi izvršavanja	3
1.3.2	Grafici ubrzanja	6
1.3.3	Diskusija dobijenih rezultata	7
2	PROBLEM 2 - FEYMAN	8
2.1	Tekst problema	8
2.2	Delovi koje treba paralelizovati	8
2.2.1	Diskusija	8
2.2.2	Način paralelizacije	8
2.3	Rezultati	9
2.3.1	Logovi izvršavanja	9
2.3.2	Grafici ubrzanja	11
2.3.3	Diskusija dobijenih rezultata	12
3	PROBLEM 3 - MOLDYN	13
3.1	Tekst problema	13
3.2	Delovi koje treba paralelizovati	13
3.2.1	Diskusija	13
3.2.2	Način paralelizacije	13
3.3	Rezultati	14
3.3.1	Logovi izvršavanja	14
3.3.2	Grafici ubrzanja	15
3.3.3	Diskusija dobijenih rezultata	15
4	PROBLEM 4 - MOLDYN	16
4.1	Tekst problema	16
4.2	Delovi koje treba paralelizovati	16
4.2.1	Diskusija	16
4.2.2	Način paralelizacije	16
4.3	Rezultati	17
4.3.1	Logovi izvršavanja	17
4.3.2	Grafici ubrzanja	19
4.3.3	Diskusija dobijenih rezultata	19

1 PROBLEM 1 - PRIME

1.1 Tekst problema

Paralelizovati program koji vrši određivanje ukupnog broja prostih brojeva u zadatom opsegu. Program se nalazi u datoteci **prime.c** u arhivi koja je priložena uz ovaj dokument. Proces sa rangom 0 treba da učitava ulazne podatke, raspodeli posao ostalim procesima, na kraju prikupi dobijene rezultate i ravnopravno učestvuje u obradi. Za razmenu podataka, koristiti rutine za kolektivnu komunikaciju. Program testirati sa parametrima koji su dati u datoteci **run**. [1, N]

1.2 Delovi koje treba paralelizovati

1.2.1 Diskusija

Delovi koda, koje je moguće paralelizovati, predstavljaju dve for petlje, od kojih je jedna ugnježdjena i while petlja. Paralelizovana je spoljna for petlja, koja iterira kroz brojeve u zadatom opsegu. Unutrašnja for petlja nije paralelizovana, a njen zadatak je da proveri da li je trenutno posmatrani broj iz opsega deljiv nekim od brojeva manjih od njega. Razlog za ovo je što se prednost uvek daje paralelizaciji spoljne petlje. While petlja se nalazi u delu koda koji nije eksplicitno deo algoritma, već pripada delu koji se odnosi na testiranje rada programa, zbog ovoga nije rađena paralelizacija.

1.2.2 Način paralelizacije

Master je zadužen za učitavanje i ispis vrednosti, pa tako na početku samo on učitava parametre i radi broadcast vrednosti ostalim procesima. Zatim svaki proces na osnovu svog ranga u okviru komunikatora računa koji opseg brojeva treba da obradi. Takođe, svaki proces izračunava svoju total promenljivu, pa kada se završi celokupna obrada na svim procesima radi se redukcija date promenljive po operaciji zbira kod master procesa.

Isprobana su dva načina podele opsega brojeva na procese, a to su blokovska i ciklična raspodela, gde je varirana veličina bloka za cikličnu raspodelu.

1.3 Rezultati

1.3.1 Logovi izvršavanja

```
Running variant: prime_v0seq
Running test case: 1 of 3 (1 65536 4) N=1 3 times
  AVG Time: 0.2735
Running test case: 2 of 3 (1 131072 2) N=1 3 times
  AVG Time: 1.3006
Running test case: 3 of 3 (5 500000 10) N=1 3 times
  AVG Time: 12.5617
Running variant: prime_v1blk
Running test case: 1 of 3 (1 65536 4) N=2 3 times
  AVG Time: 0.2005 AVG Speedup: 1.36 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 3 times
  AVG Time: 0.1182 AVG Speedup: 2.31 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=2 3 times
  AVG Time: 0.9430 AVG Speedup: 1.38 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 3 times
  AVG Time: 0.5619 AVG Speedup: 2.31 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=2 3 times
  AVG Time: 9.1251 AVG Speedup: 1.38 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 3 times
  AVG Time: 5.4555 AVG Speedup: 2.30 ALL Tests PASSED
Running variant: prime_v2cyc
Running test case: 1 of 3 (1 65536 4) N=2 3 times
  AVG Time: 0.2687 AVG Speedup: 1.02 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 3 times
  AVG Time: 0.1403 AVG Speedup: 1.95 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=2 3 times
  AVG Time: 1.2852 AVG Speedup: 1.01 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 3 times
  AVG Time: 0.6685 AVG Speedup: 1.95 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=2 3 times
  AVG Time: 12.4098 AVG Speedup: 1.01 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 3 times
  AVG Time: 6.4354 AVG Speedup: 1.95 ALL Tests PASSED
Running variant: prime_v3adv
Running test case: 1 of 3 (1 65536 4) N=2 3 times
  AVG Time: 0.1373 AVG Speedup: 1.99 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 3 times
  AVG Time: 0.0707 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=2 3 times
  AVG Time: 0.6582 AVG Speedup: 1.98 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 3 times
  AVG Time: 0.3385 AVG Speedup: 3.84 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=2 3 times
  AVG Time: 6.2966 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 3 times
  AVG Time: 3.2192 AVG Speedup: 3.90 ALL Tests PASSED
```

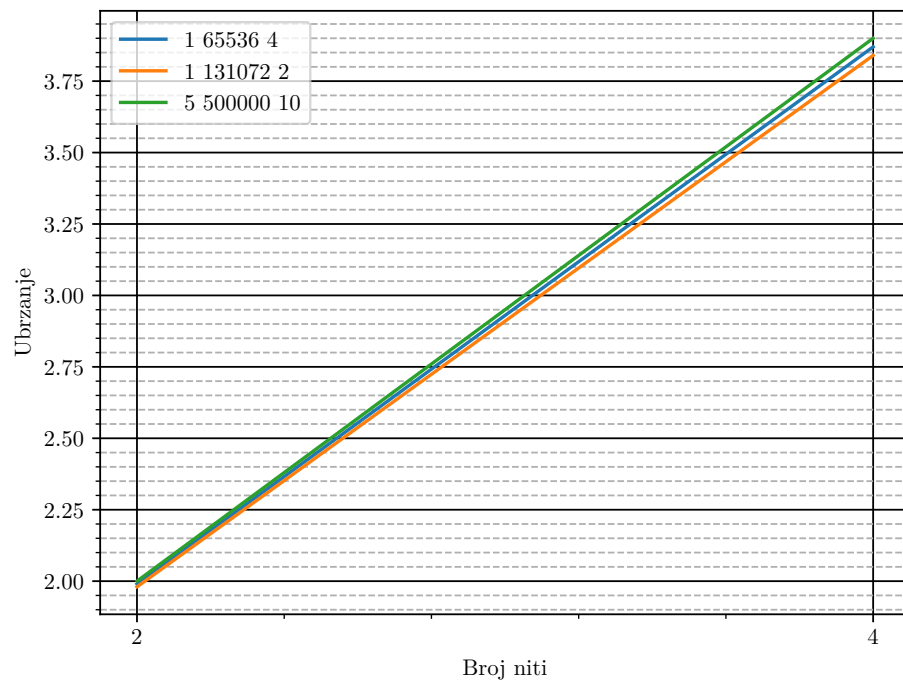
```

Running task: dz2z1
Running variant: prime_v0seq
Running test case: 1 of 3 (1 65536 4) N=1 2 times
    AVG Time: 0.2735
Running test case: 2 of 3 (1 131072 2) N=1 2 times
    AVG Time: 1.3005
Running test case: 3 of 3 (5 500000 10) N=1 2 times
    AVG Time: 12.5616
Running variant: prime_v4adv_t0
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 1 2 times
    AVG Time: 0.1404 AVG Speedup: 1.95 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 2 2 times
    AVG Time: 0.0709 AVG Speedup: 3.86 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 4 2 times
    AVG Time: 0.0704 AVG Speedup: 3.89 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 8 2 times
    AVG Time: 0.0709 AVG Speedup: 3.86 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 16 2 times
    AVG Time: 0.0707 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 32 2 times
    AVG Time: 0.0713 AVG Speedup: 3.83 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 64 2 times
    AVG Time: 0.0708 AVG Speedup: 3.86 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 128 2 times
    AVG Time: 0.0716 AVG Speedup: 3.82 ALL Tests PASSED
Running test case: 1 of 3 (1 65536 4) N=4 parameter: 256 2 times
    AVG Time: 0.0720 AVG Speedup: 3.80 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 1 2 times
    AVG Time: 0.6687 AVG Speedup: 1.94 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 2 2 times
    AVG Time: 0.3378 AVG Speedup: 3.85 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 4 2 times
    AVG Time: 0.3378 AVG Speedup: 3.85 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 8 2 times
    AVG Time: 0.3364 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 16 2 times
    AVG Time: 0.3381 AVG Speedup: 3.85 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 32 2 times
    AVG Time: 0.3364 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 64 2 times
    AVG Time: 0.3364 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 128 2 times
    AVG Time: 0.3366 AVG Speedup: 3.86 ALL Tests PASSED
Running test case: 2 of 3 (1 131072 2) N=4 parameter: 256 2 times
    AVG Time: 0.3363 AVG Speedup: 3.87 ALL Tests PASSED

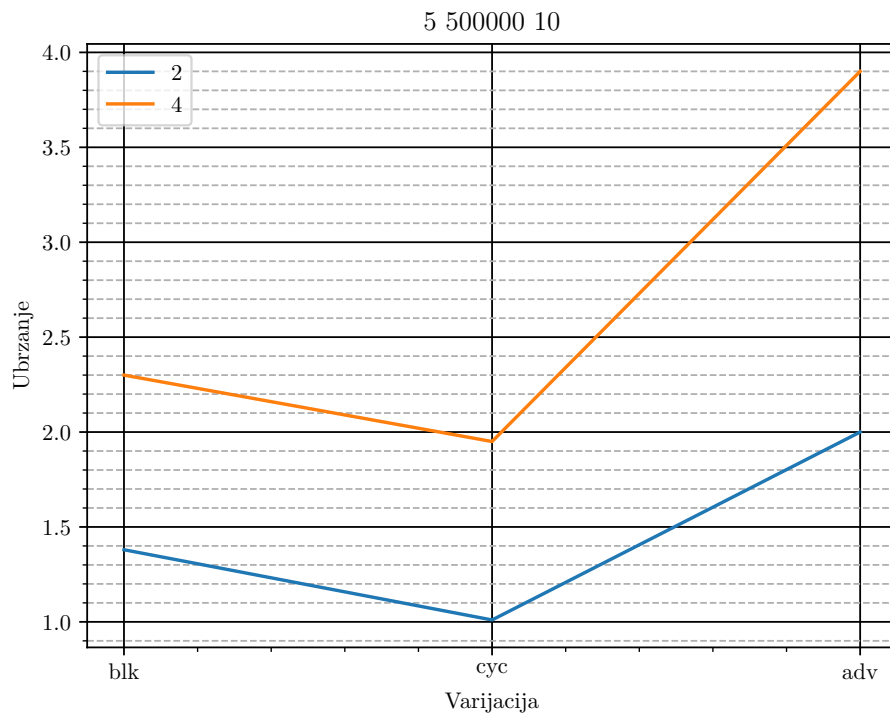
```

Running test case: 3 of 3 (5 500000 10) N=4 parameter: 1 2 times
AVG Time: 6.4313 AVG Speedup: 1.95 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 2 2 times
AVG Time: 3.2280 AVG Speedup: 3.89 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 4 2 times
AVG Time: 3.2329 AVG Speedup: 3.89 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 8 2 times
AVG Time: 3.2202 AVG Speedup: 3.90 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 16 2 times
AVG Time: 3.2188 AVG Speedup: 3.90 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 32 2 times
AVG Time: 3.2245 AVG Speedup: 3.90 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 64 2 times
AVG Time: 3.2448 AVG Speedup: 3.87 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 128 2 times
AVG Time: 3.2295 AVG Speedup: 3.89 ALL Tests PASSED
Running test case: 3 of 3 (5 500000 10) N=4 parameter: 256 2 times
AVG Time: 3.2324 AVG Speedup: 3.89 ALL Tests PASSED

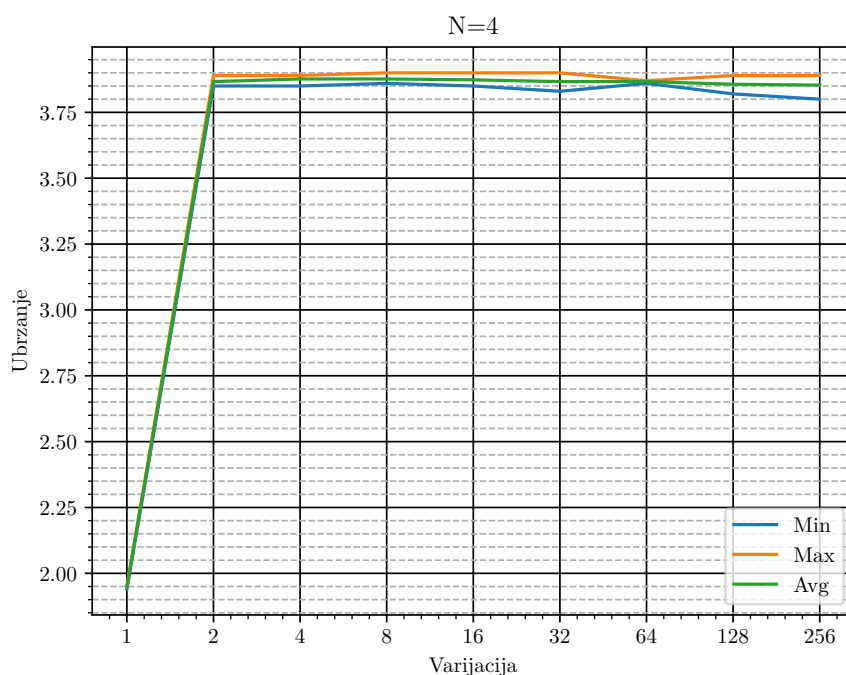
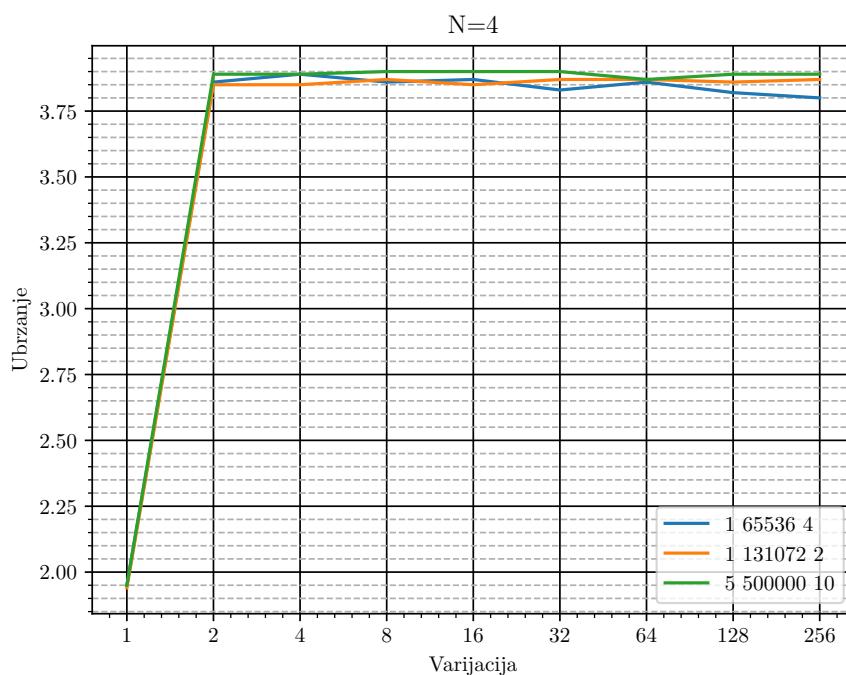
1.3.2 Grafici ubrzanja



Slika 1: Grafik zavisnosti ubrzanja od broja niti za različite test primere



Slika 2: Grafik zavisnosti ubrzanja od varijacije za različit broj niti



Slika 3: Grafik zavisnosti ubrzanja od veličine bloka

1.3.3 Diskusija dobijenih rezultata

Najbolje performanse su dobijene za cikličnu podelu sa veličinom bloka 16, što je posledica raspodele prostih brojeva. Kako je najveća koncentracija na početku opsega na ovaj način sve niti dobiju po deo ovog intervala, pa je podela posla ravnomerna. Ukoliko je veličina bloka jedan dve niti dobijaju samo parne brojeve na obradu i samim tim nijedan broj koji obrađuju nije prost, raspodela posla u ovom slučaju je veoma neravnomerna što negativno utiče na performanse. U slučaju blokovske podele, posao je nešto ravnomerniji, ali problem predstavlja što prva nit dobija blok sa najvećom koncentracijom prostih brojeva.

2 PROBLEM 2 - FEYMAN

2.1 Tekst problema

Paralelizovati program koji vrši izračunavanje 3D Poasonove jednačine korišćenjem Feyman-Kac algoritma. Algoritam stohastički računa rešenje parcijalne diferencijalne jednačine krenuvši N puta iz različitih tačaka domena. Tačke se kreću po nasumičnim putanjama i prilikom izlaska iz granica domena kretanje se zaustavlja računajući dužinu puta do izlaska. Proces se ponavlja za svih N tačaka i konačno aproksimira rešenje jednačine. Program se nalazi u datoteci **feyman.c** u arhivi koja je priložena uz ovaj dokument. Program testirati sa parametrima koji su dati u datoteci **run**. [1, N]

2.2 Delovi koje treba paralelizovati

2.2.1 Diskusija

Deo programa, koji je moguće paralelizovati predstavljaju tri for petlje, koje su kombinacijom vrednosti pretvorene u jednu ekvivalentnu petlju radi lakše podele posla. Takođe, postoji i četvrta ugnježdjena for petlja čija paralelizacija nije rađena, jer bi granulacija posla po procesu bila veoma mala.

2.2.2 Način paralelizacije

Kada je reč o paralelizaciji for petlje isprobana su dva pristupa, odnosno manager - worker model i statička podele posla. Kao i u prethodnom primeru master je zadužen za operacije ispisa.

Kod manager - worker modela master izvršava for petlju i ukoliko se ne izvrši naredba continue šalje koordinate radnicima na obradu, pri čemu čeka da mu se radnik javi da je spreman. Na početku se svi radnici jave menadžeru, a zatim se u petlji javljaju kada završe obradu sa izračunatom promenljivom err i čekaju narednu poruku. Master po prijemu poruke od radnika uzima izračunat err i dodaje na ukupnu vrednost. Kada koordinata za slanje više nema, master sačeka odgovore od uposlenih radnika i obavesti sve radnike o kraju rada.

Kod statičke podele, rađen je blokovski i cikličan pristup. Svaki od procesa prema rangu pre-računa koji opseg koordinata treba da obradi. Na kraju celokupne obrade radi se redukcija za promenljive err i n_inside po operaciji zbira na master procesu.

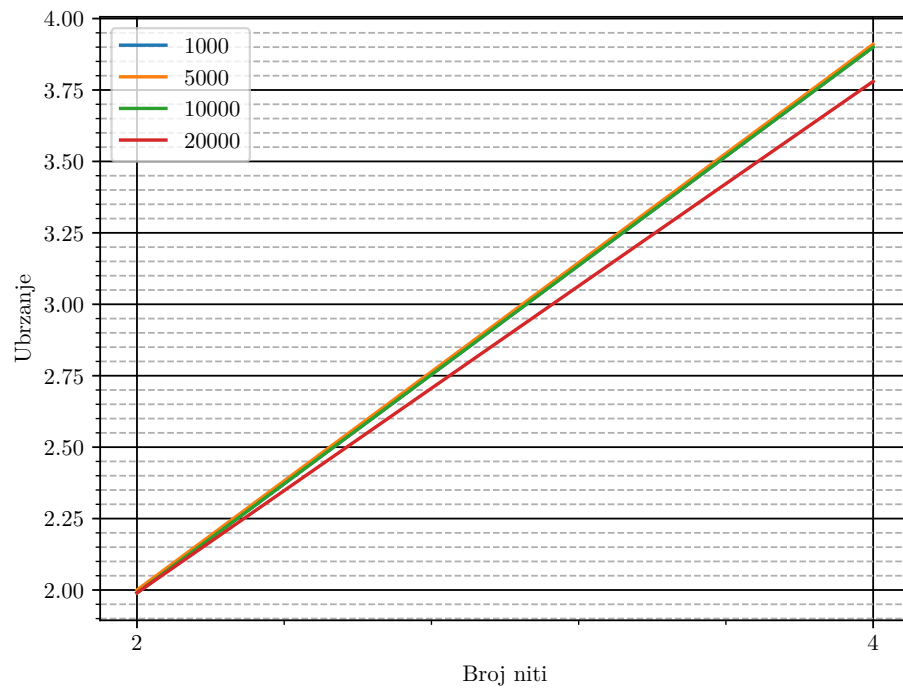
2.3 Rezultati

2.3.1 Logovi izvršavanja

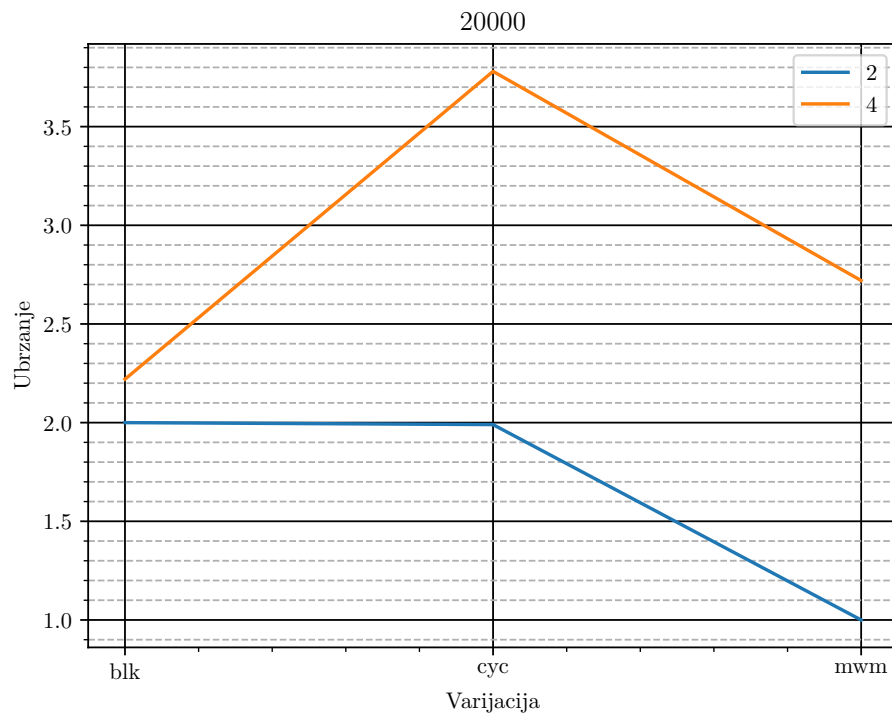
```
Running task: dz2z2
Running variant: feyman_v0seq
Running test case: 1 of 4 (1000) N=1 3 times
    AVG Time: 3.0041
Running test case: 2 of 4 (5000) N=1 3 times
    AVG Time: 15.0616
Running test case: 3 of 4 (10000) N=1 3 times
    AVG Time: 30.1144
Running test case: 4 of 4 (20000) N=1 3 times
    AVG Time: 60.2143
Running variant: feyman_v1blk
Running test case: 1 of 4 (1000) N=2 3 times
    AVG Time: 1.5027 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 1 of 4 (1000) N=4 3 times
    AVG Time: 1.2666 AVG Speedup: 2.37 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=2 3 times
    AVG Time: 7.5174 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=4 3 times
    AVG Time: 6.7276 AVG Speedup: 2.24 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=2 3 times
    AVG Time: 15.0375 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=4 3 times
    AVG Time: 13.5391 AVG Speedup: 2.22 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=2 3 times
    AVG Time: 30.0844 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=4 3 times
    AVG Time: 27.0953 AVG Speedup: 2.22 ALL Tests PASSED
Running variant: feyman_v2cyc
Running test case: 1 of 4 (1000) N=2 3 times
    AVG Time: 1.5119 AVG Speedup: 1.99 ALL Tests PASSED
Running test case: 1 of 4 (1000) N=4 3 times
    AVG Time: 0.7702 AVG Speedup: 3.90 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=2 3 times
    AVG Time: 7.5437 AVG Speedup: 2.00 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=4 3 times
    AVG Time: 3.8549 AVG Speedup: 3.91 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=2 3 times
    AVG Time: 15.1076 AVG Speedup: 1.99 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=4 3 times
    AVG Time: 7.7132 AVG Speedup: 3.90 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=2 3 times
    AVG Time: 30.1834 AVG Speedup: 1.99 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=4 3 times
    AVG Time: 15.9344 AVG Speedup: 3.78 ALL Tests PASSED
```

```
Running variant: feyman_v3mwm
Running test case: 1 of 4 (1000) N=2 3 times
    AVG Time: 3.0194 AVG Speedup: 0.99 ALL Tests PASSED
Running test case: 1 of 4 (1000) N=4 3 times
    AVG Time: 1.0291 AVG Speedup: 2.92 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=2 3 times
    AVG Time: 15.1341 AVG Speedup: 1.00 ALL Tests PASSED
Running test case: 2 of 4 (5000) N=4 3 times
    AVG Time: 5.1508 AVG Speedup: 2.92 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=2 3 times
    AVG Time: 30.2711 AVG Speedup: 0.99 ALL Tests PASSED
Running test case: 3 of 4 (10000) N=4 3 times
    AVG Time: 10.3893 AVG Speedup: 2.90 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=2 3 times
    AVG Time: 60.5103 AVG Speedup: 1.00 ALL Tests PASSED
Running test case: 4 of 4 (20000) N=4 3 times
    AVG Time: 22.1730 AVG Speedup: 2.72 ALL Tests PASSED
```

2.3.2 Grafici ubrzanja



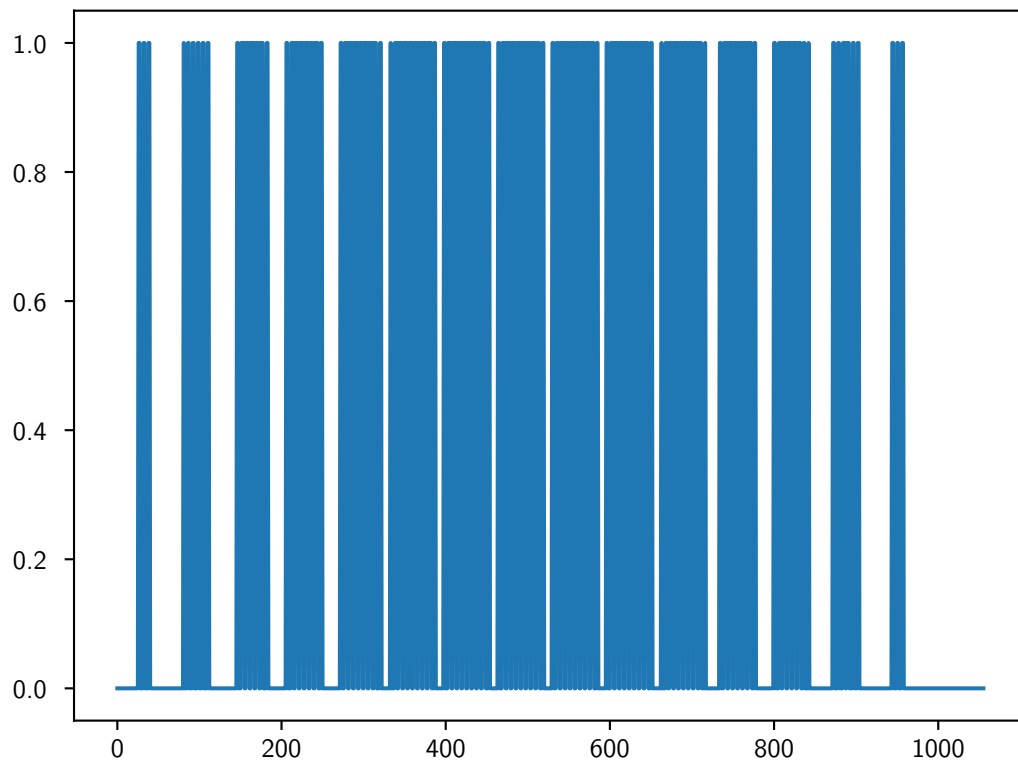
Slika 4: Grafik zavisnosti ubrzanja od broja niti za različite test primere



Slika 5: Grafik zavisnosti ubrzanja od varijacije za različit broj niti

2.3.3 Diskusija dobijenih rezultata

Prvo što se može primetiti jeste da manager - worker model ne daje dobre performanse jer je režijski trošak komunikacije za rešenje ovog problema suviše velik, pa ima značajan uticaj na brzinu izvršavanja. Kada se posmatra ukupno ponašanje programa za 2 i 4 niti ispostavlja se da je ciklična podela posla najbolja za ovaj problem, dok blokovska značajno varira sa odličnim performansama za 2 niti i najgorim za 4. Uticaj na ovo ima sama priroda problema, koji algoritam obrađuje.



Slika 6: Grafik raspodele iteracija koje izvršavaju unutrašnju petlju

Dobre performanse blokovske raspodele za 2 niti objašnjava to što se uočava simetrija na prethodnom grafiku, dok za 4 velika koncentracija preskakanja continue naredbe na sredini opsega izvršavanja uz simetriju dovodi do veoma loših performansi, jer će dve niti imati nesrazmerno veliki posao u odnosu na ostale.

3 PROBLEM 3 - MOLDYN

3.1 Tekst problema

Paralelizovati jednostavan program koji se bavi molekularnom dinamikom. Kod predstavlja simulaciju molekularne dinamike argonovog atoma u ograničenom prozoru (prostoru) sa periodičnim graničnim uslovima. Atomi se inicijalno nalaze raspoređeni u pravilnu mrežu, a zatim se tokom simulacije dešavaju interakcije između njih. U svakom koraku simulacije u glavnoj petlji se dešava sledeće:

- Čestice (atomi) se pomeraju zavisno od njihovih brzina i brzine se parcijalno ažuriraju u pozivu funkcije **domove**.
- Sile koje se primenjuju na nove pozicije čestica se izračunavaju; takođe, akumuliraju se prosečna kinetička energija (*virial*) i potencijalna energija u pozivu funkcije **forces**.
- Sile se skaliraju, završava ažuriranje brzine i izračunavanje kinetičke energije u pozivu funkcije **mkekin**.
- Prosečna brzina čestice se računa i skaliraju temperature u pozivu funkcije **velavg**.
- Pune potencijalne i prosečne kinetičke energije (*virial*) se računaju i ispisuju u funkciji **prnout**.

Program se nalazi u direktorijumu **Moldyn** u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih su od interesa datoteke **main.c** i **forces.c**, jer se u njima provodi najviše vremena. Analizirati dati kod i obratiti pažnju na redukcione promenljive unutar datoteke **forces.c**. Ukoliko je potrebno međusobno isključenje prilikom paralelizacije programa, koristiti kritične sekcije ili atomske operacije. [1, N]

3.2 Delovi koje treba paralelizovati

3.2.1 Diskusija

U ovom zadatku za paralelizaciju su posmatrane samo funkcije **main** i **forces**. U **main** funkciji primećena je jedna **for** petlja, koja se ne može paralelizovati zbog zavisnosti po podacima. U **forces** funkciji primećene su dve ugnježdene **for** petlje. Kombinacija dve petlje nije moguća, pa je doneta odluka da se paralelizuje samo spoljna petlja.

3.2.2 Način paralelizacije

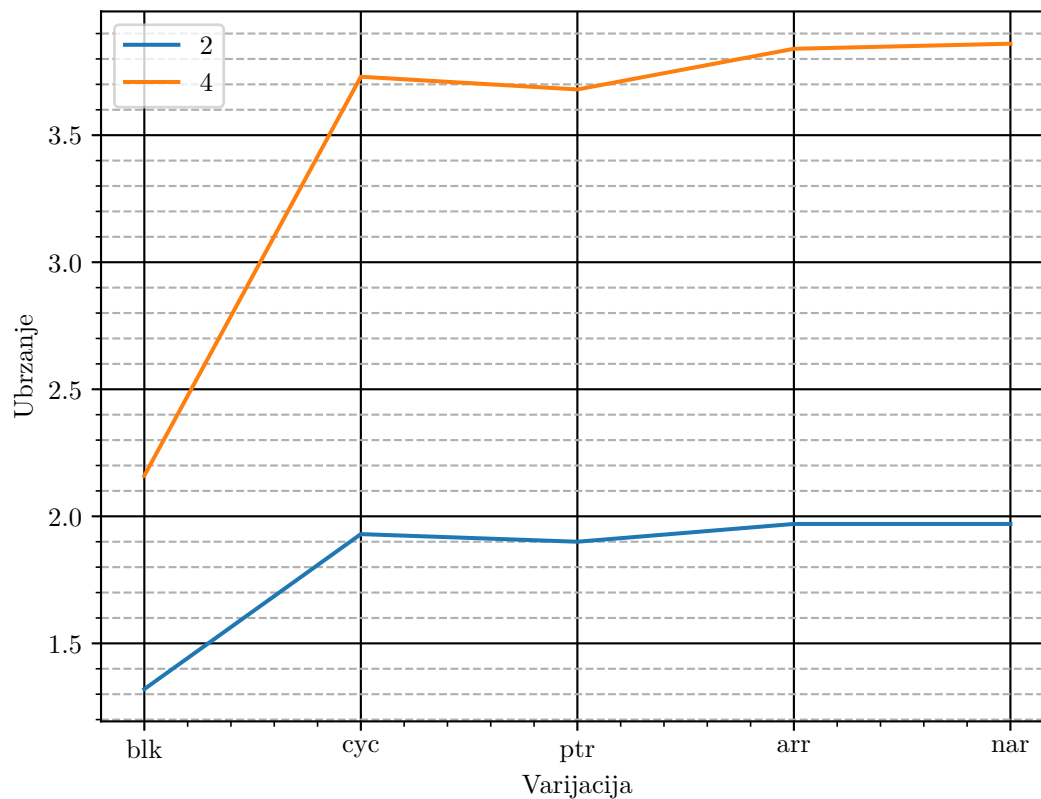
Paralelizacija petlje u **forces** funkciji je rađena primenom blokovske i ciklične raspodele posla, gde svaki proces preračuna prema rangu koji mu opseg podataka pripada za obradu. U **main** funkciji svi procesi izvršavaju sve, a redukcija se radi nad nizom **f** i promenljivama **vir** i **epot**. Pri čemu se **allReduce** koristi za niz **f** jer će svi procesi koristiti ove vrednosti. Isprobana alternativa je da master radi funkcije u **main-u**, procesi dele obradu nad **forces** funkcijom što zahteva **broadcast** niza **x** i resetovanje niza **f** sa vrednošću 0. Takođe, kako **source** i **destination** baferi moraju biti različiti objekti, radi dalje optimizacije pokušana je i zamena kopiranja uvođenje pokazivača na nizove i njihova zamena. Takođe, kako se **f** resetuje u varijanti gde samo master izvršava telo petlje osim funkcije **forces** nije potrebna upotreba **allReduce** već je obična redukcija dovoljna.

3.3 Rezultati

3.3.1 Logovi izvršavanja

```
Running task: dz2z3
Running variant: md_v0seq
Running test case: 1 of 1 () N=1 3 times
    AVG Time: 5.7964
Running variant: md_v1blk
Running test case: 1 of 1 () N=2 3 times
    AVG Time: 4.4061 AVG Speedup: 1.32 ALL Tests PASSED
Running test case: 1 of 1 () N=4 3 times
    AVG Time: 2.6798 AVG Speedup: 2.16 ALL Tests PASSED
Running variant: md_v2cyc
Running test case: 1 of 1 () N=2 3 times
    AVG Time: 3.0051 AVG Speedup: 1.93 ALL Tests PASSED
Running test case: 1 of 1 () N=4 3 times
    AVG Time: 1.5530 AVG Speedup: 3.73 ALL Tests PASSED
Running variant: md_v3ptr
Running test case: 1 of 1 () N=2 3 times
    AVG Time: 3.0566 AVG Speedup: 1.90 ALL Tests PASSED
Running test case: 1 of 1 () N=4 3 times
    AVG Time: 1.5735 AVG Speedup: 3.68 ALL Tests PASSED
Running variant: md_v4arr
Running test case: 1 of 1 () N=2 3 times
    AVG Time: 2.9465 AVG Speedup: 1.97 ALL Tests PASSED
Running test case: 1 of 1 () N=4 3 times
    AVG Time: 1.5090 AVG Speedup: 3.84 ALL Tests PASSED
Running variant: md_v5nar
Running test case: 1 of 1 () N=2 3 times
    AVG Time: 2.9498 AVG Speedup: 1.97 ALL Tests PASSED
Running test case: 1 of 1 () N=4 3 times
    AVG Time: 1.5023 AVG Speedup: 3.86 ALL Tests PASSED
```

3.3.2 Grafici ubrzanja



Slika 7: Grafik zavisnosti ubrzanja od varijacije za različit broj niti

3.3.3 Diskusija dobijenih rezultata

Kako postoji disbalans u opterećenju pri izvršavanju iteracija algoritma bolje performanse daje ciklična podela posla. Zbog ovoga ona je podrazumevana u svim daljim razmatranjima.

Mali dobitak na performansama se dobija ukoliko se jedino funkcija `forces` deli između niti, što odgovara tome da vreme izvršavanja funkcija od sinhronizacije procesa posle i pre `forces` funkcije zavisi samo od master procesa, a ne od najsporijeg. Takodje, uklanjanje korišćenja funkcije `allReduce` u korist obične redukcije i samim tim izbegavanjem kopiranja niza `f` postiže se dodatno malo ubrzanje.

Interesantno je da korišćenje pokazivača, kako bi se izbeglo kopiranje niza dovodi do pogoršanja performansi.

4 PROBLEM 4 - MOLDYN

4.1 Tekst problema

Prethodni program paralelizovati korišćenjem *manager* - *worker* modela. Proces gospodar (master) treba da učitava neophodne podatke, generiše poslove, deli posao ostalim procesima i ispiše na kraju dobijeni rezultat. U svakom koraku obrade, proces gospodar šalje procesu radniku na obradu jednu jedinicu posla čiji veličinu treba pažljivo odabrati. Proces radnik prima podatke, vrši obradu, vraća rezultat, signalizira gospodaru kada je spreman da primi sledeći posao i ponavlja opisani postupak dok ne dobije signal da prekine sa radom. Veličinu jedne jedinice posla prilagoditi karakteristikama programa. Ukoliko je moguće, koristiti rutine za neblokirajuću komunikaciju za razmenu poruka. Način pokretanja programa se nalazi u datoteci **run**. [1, N]

4.2 Delovi koje treba paralelizovati

4.2.1 Diskusija

Kao i u prethodnom zadatku za paralelizaciju su posmatrane samo funkcije *main* i *forces*. U *main* funkciji primećena je jedna *for* petlja, koja se ne može paralelizovati zbog zavisnosti po podacima. U *forces* funkciji primećene su dve ugnježdene *for* petlje. Kombinacija dve petlje nije moguća, pa je doneta odluka da se paralelizuje samo spoljna petlja.

4.2.2 Način paralelizacije

Master je zadužen za izvršavanje petlje u *main* funkciji, osim funkcije *forces* u čemu učestvuju i procesi radnici. Master radi *broadcast* niza *x* i u petlji šalje blokove na obradu radnicima, pri čemu čeka da se radnik javi da je spreman. Radnici vrše obradu dela podataka koji im master zada i vraćaju izračunate vrednosti *vir* i *epot*. Master sabira dobijene vrednosti i kada više ne bude blokova za obradu čeka rezultate od uposlenih radnika i šalje poruke svima o kraju obrade. Nakon izlaska iz funkcije *forces* radi se redukcija niza *f*.

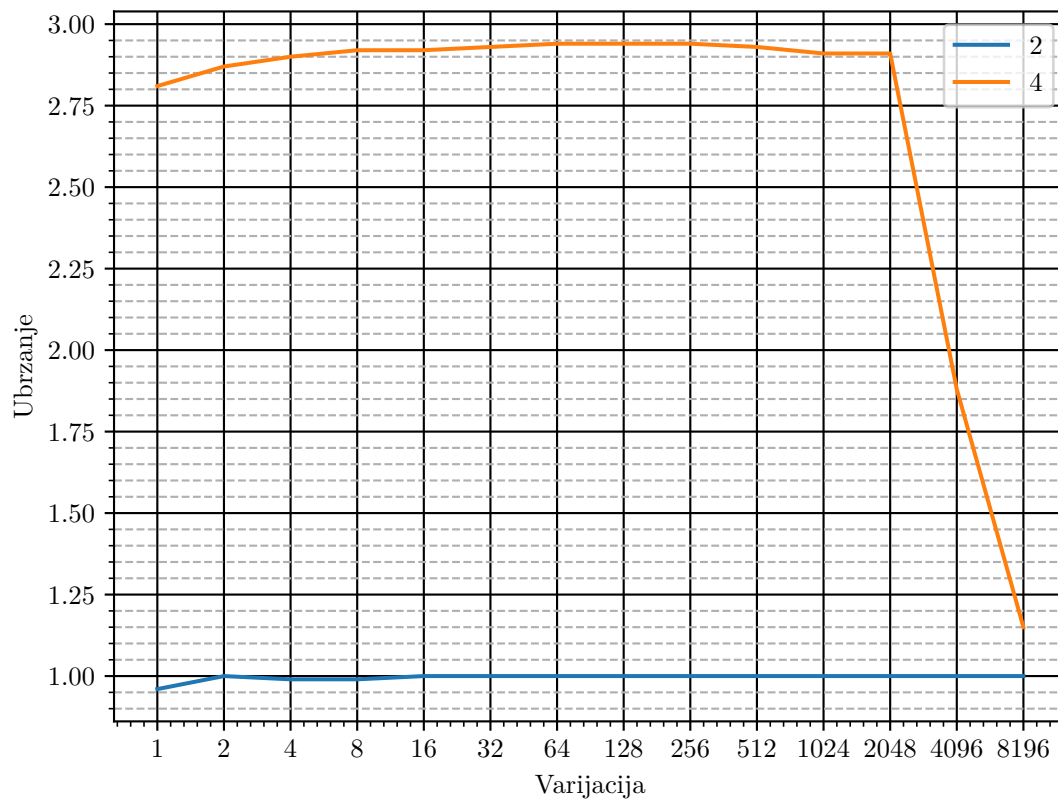
4.3 Rezultati

4.3.1 Logovi izvršavanja

```
Running task: dz2z4
Running variant: md_v0seq
Running test case: 1 of 1 () N=1 2 times
  AVG Time: 5.7966
Running variant: md_v1par_t0
Running parametrized test case: 1 of 1 () N=2 parameter: 1 2 times
  AVG Time: 6.0118 AVG Speedup: 0.96 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 2 2 times
  AVG Time: 5.8088 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 4 2 times
  AVG Time: 5.8457 AVG Speedup: 0.99 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 8 2 times
  AVG Time: 5.8289 AVG Speedup: 0.99 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 16 2 times
  AVG Time: 5.7983 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 32 2 times
  AVG Time: 5.7746 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 64 2 times
  AVG Time: 5.7845 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 128 2 times
  AVG Time: 5.7938 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 256 2 times
  AVG Time: 5.8020 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 512 2 times
  AVG Time: 5.7988 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 1024 2 times
  AVG Time: 5.8039 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 2048 2 times
  AVG Time: 5.7845 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 4096 2 times
  AVG Time: 5.7971 AVG Speedup: 1.00 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=2 parameter: 8196 2 times
  AVG Time: 5.7882 AVG Speedup: 1.00 ALL Tests PASSED
```

Running parametrized test case: 1 of 1 () N=4 parameter: 1 2 times
AVG Time: 2.0623 AVG Speedup: 2.81 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 2 2 times
AVG Time: 2.0182 AVG Speedup: 2.87 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 4 2 times
AVG Time: 2.0003 AVG Speedup: 2.90 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 8 2 times
AVG Time: 1.9857 AVG Speedup: 2.92 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 16 2 times
AVG Time: 1.9855 AVG Speedup: 2.92 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 32 2 times
AVG Time: 1.9790 AVG Speedup: 2.93 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 64 2 times
AVG Time: 1.9746 AVG Speedup: 2.94 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 128 2 times
AVG Time: 1.9726 AVG Speedup: 2.94 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 256 2 times
AVG Time: 1.9743 AVG Speedup: 2.94 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 512 2 times
AVG Time: 1.9806 AVG Speedup: 2.93 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 1024 2 times
AVG Time: 1.9945 AVG Speedup: 2.91 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 2048 2 times
AVG Time: 1.9899 AVG Speedup: 2.91 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 4096 2 times
AVG Time: 3.0845 AVG Speedup: 1.88 ALL Tests PASSED
Running parametrized test case: 1 of 1 () N=4 parameter: 8196 2 times
AVG Time: 5.0337 AVG Speedup: 1.15 ALL Tests PASSED

4.3.2 Grafici ubrzanja



Slika 8: Grafik zavisnosti ubrzanja od veličine bloka

4.3.3 Diskusija dobijenih rezultata

Režijski trošak komunikacije utiče na maksimum ubrzanja koji se postiže. Kako bi se komunikacija smanjila što je više moguće povećava se veličina bloka koja se daje na obradu radnicima. Ne sme se uzeti ni suviše veliki blok, kako opterećenje po niti ne bi postalo neravnomerno, jer blokovi bliski veličini problema dovode do nepotpune iskorišćenosti paralelizma.