UNIVERZITET U BEOGRADU - ELEKTROTEHNIČKI FAKULTET

MULTIPROCESORKI SISTEMI (13S114MUPS, 13E114MUPS)

DOMAĆI ZADATAK 1 – OPENMP

Izveštaj o urađenom domaćem zadatku

Predmetni saradnici:

doc. dr Marko Mišić

dipl. ing. Matija Dodović

Studenti:

Petar Marković 2020/0203

Lana Stamenković 2020/0206

Beograd, april 2024.

SADRŽAJ

1. PROBLEM 1 – ARITMETIČKI BROJEVI

1.1. Tekst problema

1.1.1. Zadatak 1

Paralelizovati program koji vrši izračunavanje aritmetičkih brojeva. Pozitivan ceo broj je aritmetički ako je prosek njegovih pozitivnih delilaca takođe ceo broj. Program se nalazi u datoteci **aritmetic.c** u arhivi koja je priložena uz ovaj dokument. Obratiti pažnju na ispravno deklarisanje svih promenljivih prilikom paralelizacije. Program testirati sa parametrima koji su dati u **run** skripti.

1.1.2. Zadatak 2

Rešiti prethodni problem korišćenjem koncepta poslova (tasks). Obratiti pažnju na eventualnu potrebu za sinhronizacijom i granularnost poslova. Program testirati sa parametrima koji su dati u **run** skripti.

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

U priloženom sekvencijalnom kodu za izračunavanje aritmetičkih brojeva uočili smo dve celine koje je potencijalno moguće paralelizovati. Jedna celina je petlja u **main** funkciji programa koja iterira kroz sve prirodne brojeve, proverava da li je dati broj aritmetički i staje tek kad je nađen zadati broj aritmetičkih brojeva. Druga celina u kodu koja bi mogla biti paralelizovana jesu **for** petlje u funkciji **divisor_count_and_sum()** koje traže proste činioce zadatog broja.

1.2.2. Način paralelizacije

Početna ideja je bila da se paralelizuje traženje činilaca prirodnog broja. Problem na koji se nailazi kod ovog rešenja jeste činjenica da je nakon pronalaženja prostog činioca potrebno ispitivani broj deli datim činiocem, kako bi dati algoritam funkcionisao ispravno. To proizvodi zavisnost po podacima. Još jedan problem je to što se u svakoj iteraciji petlje u **main** funkciji iznova ulazi u paralelni region, zbog čega nismo uspeli da postignemo ubrzanje ni sa jednim pokušanim rešenjem.

Problem kod paralelizacije petlje koja za svaki prirodni broj ispituje da li je aritmetički se ogleda u logičkom uslovu, zbog čega ovo nije pravilna pravougaona petlja. Zato nije moguće deterministički odrediti broj iteracija petlje i primeniti paralelizaciju pomoću worksharing direktiva.

Ideja za paralelizaciju programa se oslanja na činjenicu da je gustina aritmetičkih brojeva približno ½. Sa sigurnošću možemo da tvrdimo da će broj aritmetičkih brojeva u nekom intervalu biti manji ili jednak broju prirodnih brojeva u tom intervalu, odnosno veličini intervala. Na osnovu toga, u prvom koraku tražimo aritmetičke brojeve u intervalu od 1 do traženog broja aritmetičkih brojeva. Dati interval može jednostavno da se podeli na podintervale koji će biti dodeljeni pojedinačnim nitima na obradu. Kada sve niti završe obradu dodeljenih intervala, sumiramo koliko je ukupno nađeno aritmetičkih brojeva. U narednom koraku, računamo razliku traženog broja i broja pronađenih aritmetičkih brojeva. Dobijena razlika predstavlja veličinu narednog intervala za pretragu. Ovaj postupak se ponavlja do trenutka kada je potrebno naći samo sledeći aritmetički broj, što će veoma brzo moći jedna nit da odradi.

1.2.3. Zadatak 1 – Paralelizacija pomoću worksharing direktiva

Ideja ove implementacije jeste da se podela intervala vrši uz pomoć **worksharing** direktiva. Potrebno je uvesti samo jednu okružujuću **while** petlju koja ispituje da li je nađeno dovoljno aritmetičkih brojeva, ili da u suprotnom sračuna veličinu narednog intervala za ispitivanje. Ukupan broj nađenih aritmetičkih brojeva je u ovom slučaju redukciona promenljiva.

1.2.4. Zadatak 2 – Paralelizacija pomoću taskova

Prilikom paralelizacije pomoću taskova, ispitivanje uslova i podelu posla vrši jedna nit unutar single bloka. Prebrojavanje aritmetičkih brojeva na podintervalima izvršavaju taskovi, koji čuvaju lokalne evidencije koliko su brojeva našli, a zatim na kraju tela taska se lokalna evidencija atomično sabira sa ukupnim brojem aritmetickih brojeva. U single bloku se čeka kraj rada svih taskova pomoću taskwait direktive, pre nego što se pređe u narednu iteraciju petlje.

1.3. Rezultati – Zadatak 1

U okviru ove sekcije su izloženi rezultati paralelizacije koda za traženje aritmetičkih brojeva pomoću worksharing direktiva.

1.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti.

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.005s
user    0m0.005s
sys     0m0.000s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.014s
user    0m0.014s
sys     0m0.000s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.176s
user    0m0.176s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m4.195s
user    0m4.191s
sys     0m0.004s
```

Listing 1. Sekvencijalno izvršavanje

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.005s
user    0m0.005s
sys     0m0.001s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.014s
user    0m0.014s
sys     0m0.001s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.178s
user    0m0.177s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m4.202s
user    0m4.202s
sys     0m0.000s
```

Listing 2. Izvršavanje paralelizovanog koda za zadatak 1 za 1 nit

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.004s
user    0m0.000s
sys     0m0.006s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.010s
user    0m0.015s
sys     0m0.004s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.102s
user    0m0.185s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m2.499s
user    0m4.199s
sys     0m0.000s
```

Listing 3. Izvršavanje paralelizovanog koda za zadatak 1 za 2 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.003s
user    0m0.001s
sys     0m0.006s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.008s
user    0m0.022s
sys     0m0.005s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.057s
user    0m0.211s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m1.335s
user    0m4.329s
sys     0m0.000s
```

Listing 4. Izvršavanje paralelizovanog koda za zadatak 1 za 4 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.003s
user    0m0.009s
sys     0m0.000s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.007s
user    0m0.034s
sys     0m0.004s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.034s
user    0m0.266s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m0.717s
user    0m4.604s
sys     0m0.000s
```

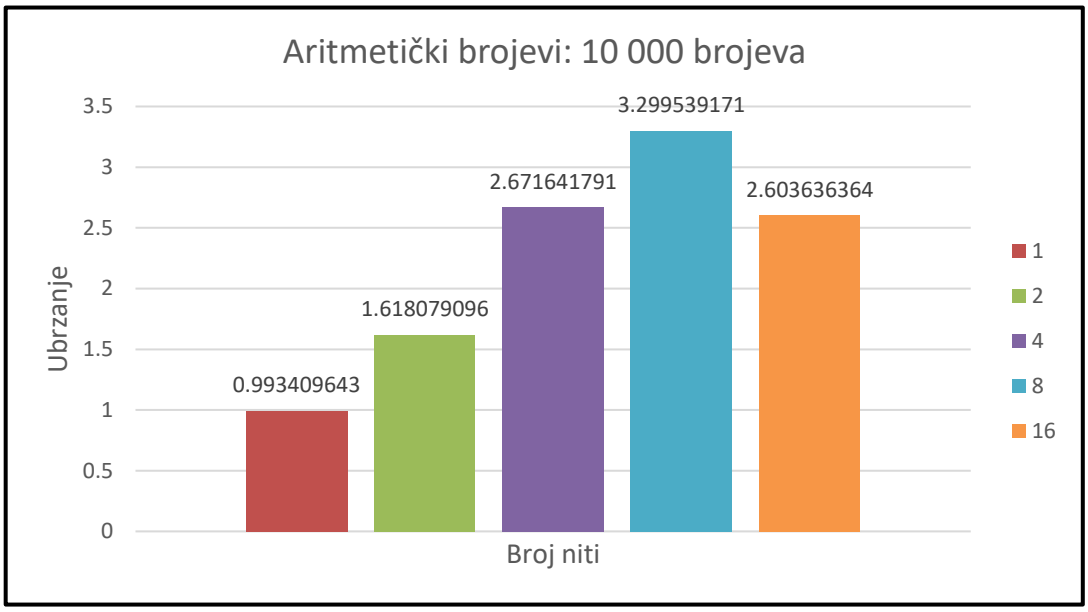
Listing 5. Izvršavanje paralelizovanog koda za zadatak 1 za 8 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.004s
user    0m0.001s
sys     0m0.018s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.008s
user    0m0.093s
sys     0m0.000s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.030s
user    0m0.466s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m0.609s
user    0m8.165s
sys     0m0.000s
```

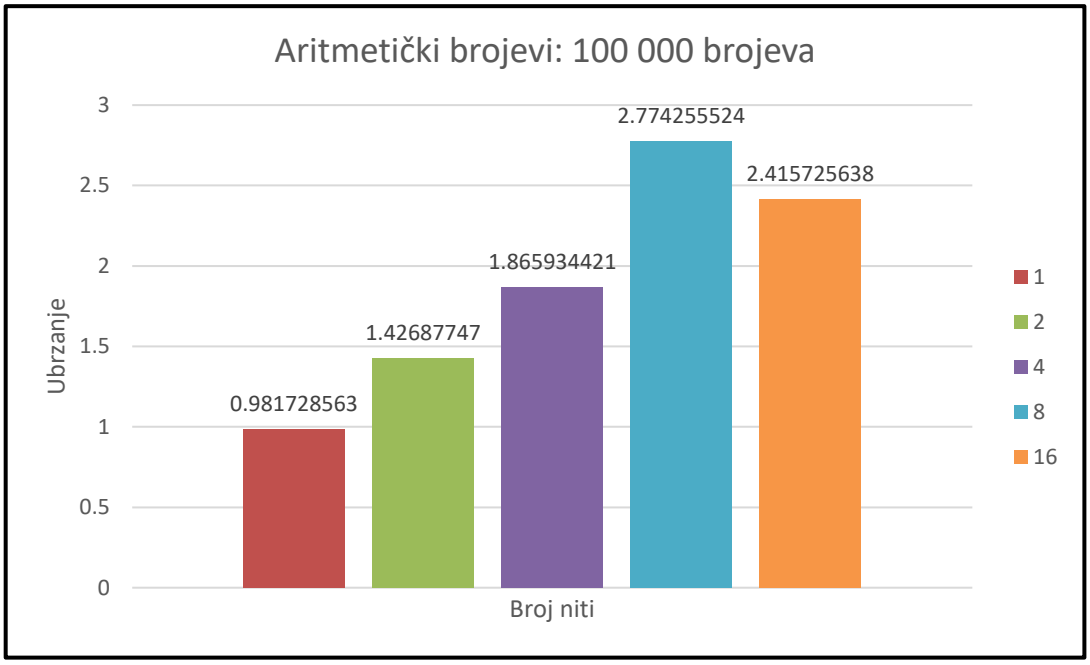
Listing 6. Izvršavanje paralelizovanog koda za zadatak 1 za 16 niti

1.3.2. Grafici ubrzanja

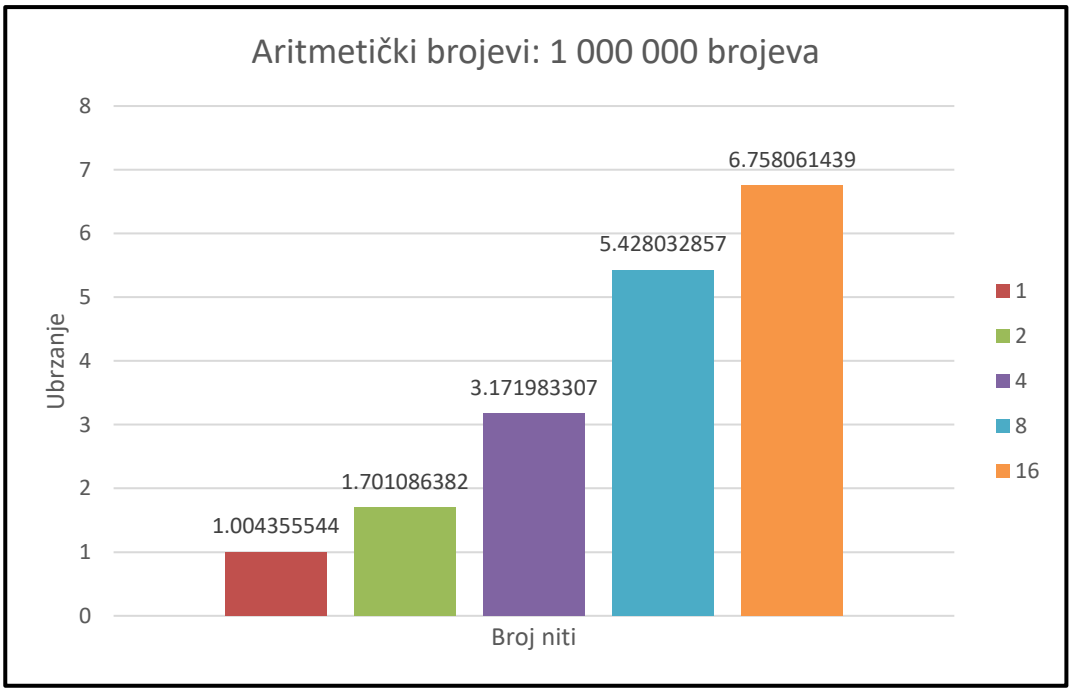
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



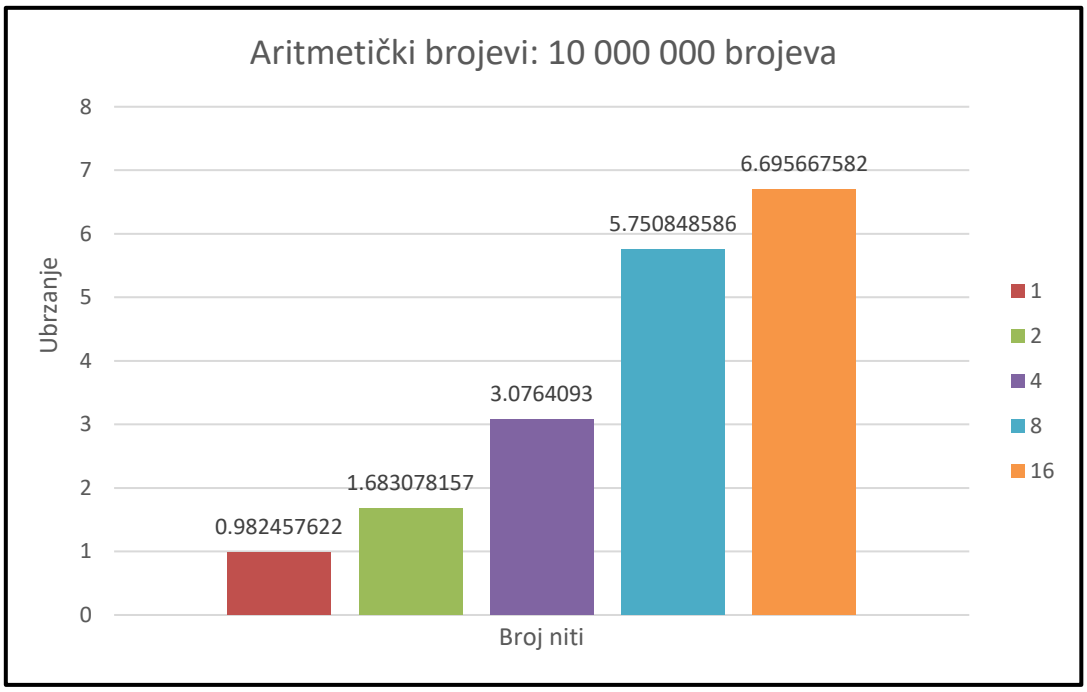
Slika 1. Aritmetički brojevi: 10 000 brojeva



Slika 2. Aritmetički brojevi: 100 000 brojeva



Slika 3. Aritmetički brojevi: 1 000 000 brojeva



Slika 4. Aritmetički brojevi: 10 000 000 brojeva

1.3.3. Diskusija dobijenih rezultata

Dobijeno rešenje ostvaruje ubrzanje za sve brojeve niti (od 2 do 16) i za sve veličine ulaznih podataka iz test primera. Bolji paralelizam se ostvaruje za veće ulazne podatke. Jedan od razloga je verovatno što se u svakoj iteraciji spoljne petlje ponovo ulazi u paralelni region. Broj iteracija spoljne petlje je $\log_2(n)$ i ovaj broj sporije raste za veće n. Dalje bi mogla da se razmatra sama podela intervala na podintervale. U trenutnoj implementaciji, ta podela je blokovska. Međutim možda bi ciklična i eksponencijalna raspodela posla mogle da daju dodatno ubrzanje.

1.4. Rezultati – Zadatak 2

U okviru ove sekcije su izloženi rezultati paralelizacije koda za traženje aritmetičkih brojeva pomoću taskova.

1.4.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti.

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.005s
user    0m0.005s
sys     0m0.000s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.014s
user    0m0.014s
sys     0m0.000s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.176s
user    0m0.176s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m4.195s
user    0m4.191s
sys     0m0.004s
```

Listing 1. Sekvencijalno izvršavanje

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.005s
user    0m0.005s
sys     0m0.001s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.014s
user    0m0.010s
sys     0m0.004s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.177s
user    0m0.177s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m4.203s
user    0m4.203s
sys     0m0.000s
```

Listing 2. Izvršavanje paralelizovanog koda za zadatak 2 za 1 nit

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.004s
user    0m0.006s
sys     0m0.001s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.007s
user    0m0.011s
sys     0m0.001s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.105s
user    0m0.189s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m2.501s
user    0m4.231s
sys     0m0.000s
```

Listing 3. Izvršavanje paralelizovanog koda za zadatak 2 za 2 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.004s
user    0m0.007s
sys     0m0.000s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.009s
user    0m0.028s
sys     0m0.000s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.056s
user    0m0.207s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m1.337s
user    0m4.331s
sys     0m0.000s
```

Listing 4. Izvršavanje paralelizovanog koda za zadatak 2 za 4 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.004s
user    0m0.000s
sys     0m0.011s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.007s
user    0m0.041s
sys     0m0.001s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.031s
user    0m0.237s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m0.717s
user    0m4.605s
sys     0m0.000s
```

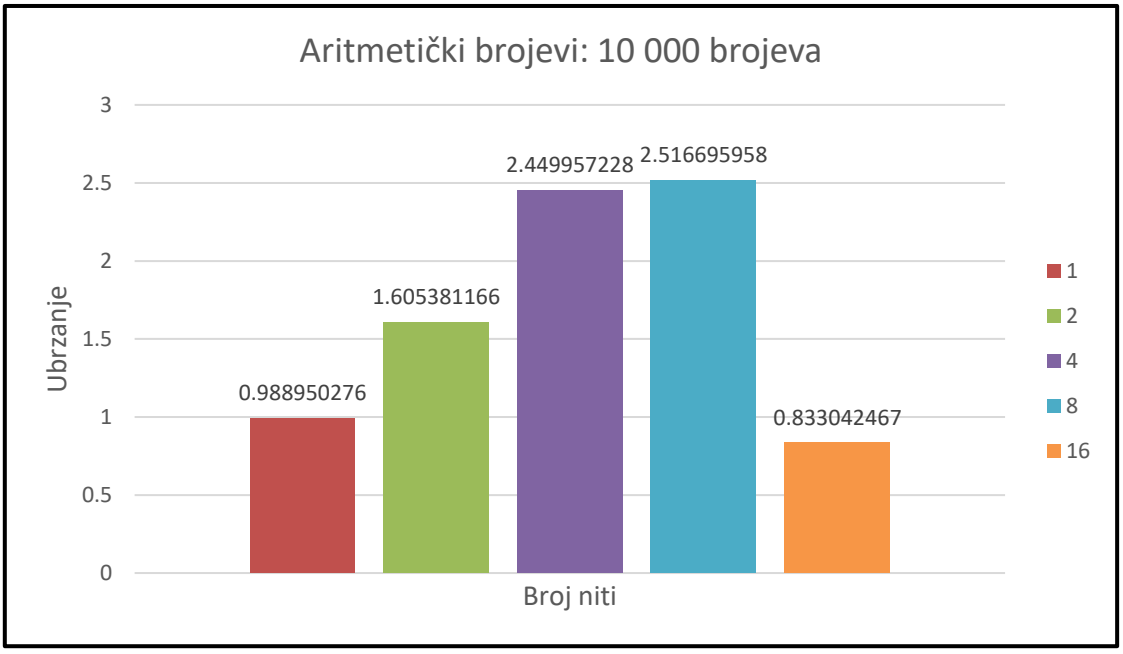
Listing 5. Izvršavanje paralelizovanog koda za zadatak 2 za 8 niti

```
10001th arithmetic number is 12955
Number of composite arithmetic numbers <= 12955: 8459

real    0m0.005s
user    0m0.040s
sys     0m0.000s

100001th arithmetic number is 125589
Number of composite arithmetic numbers <= 125589: 88220

real    0m0.007s
user    0m0.071s
sys     0m0.000s

1000001th arithmetic number is 1228665
Number of composite arithmetic numbers <= 1228665: 905044

real    0m0.028s
user    0m0.421s
sys     0m0.000s

10000001th arithmetic number is 12088245
Number of composite arithmetic numbers <= 12088245: 9206548

real    0m0.659s
user    0m8.269s
sys     0m0.000s
```

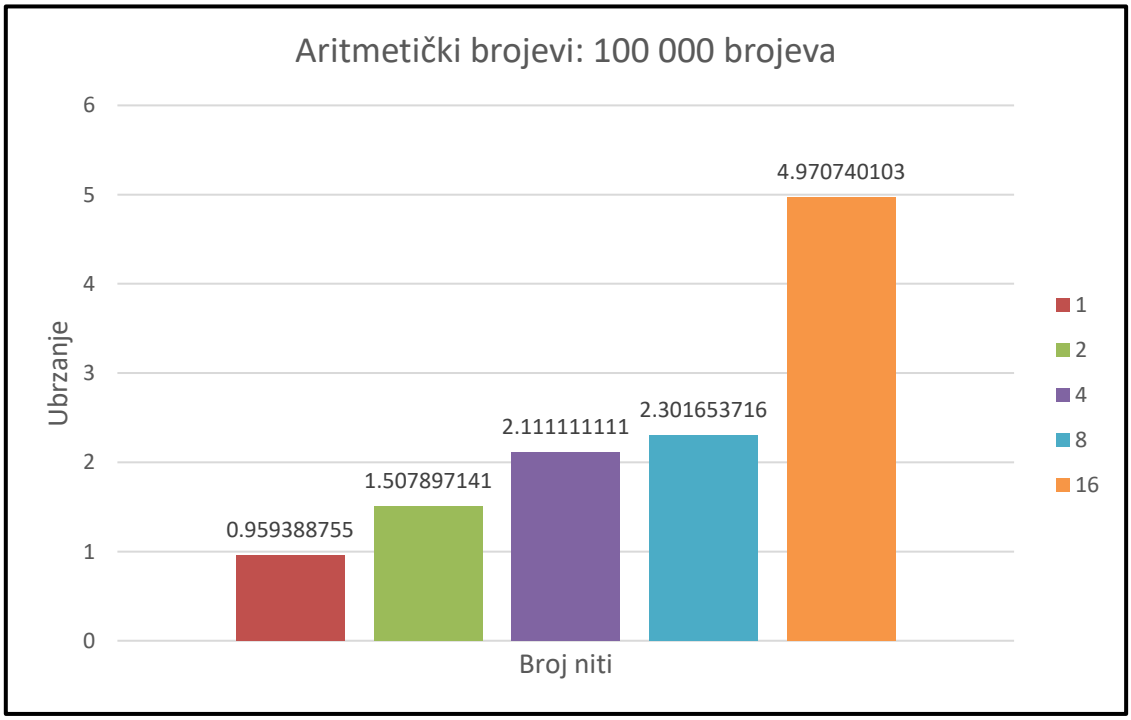
Listing 6. Izvršavanje paralelizovanog koda za zadatak 2 za 16 niti

1.4.2. Grafici ubrzanja

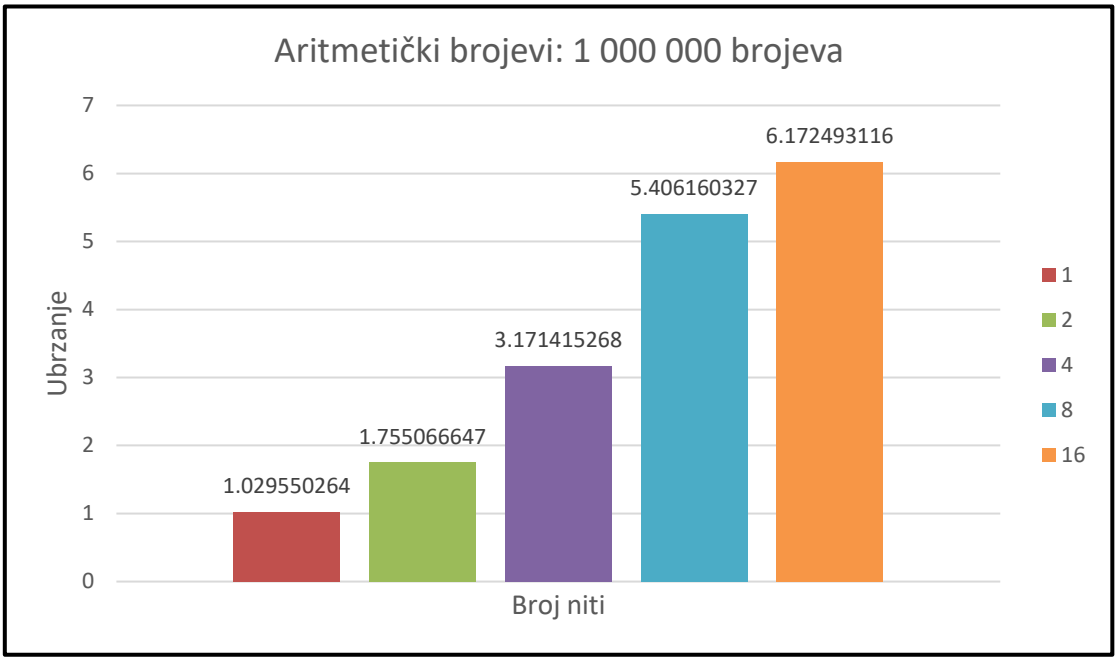
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



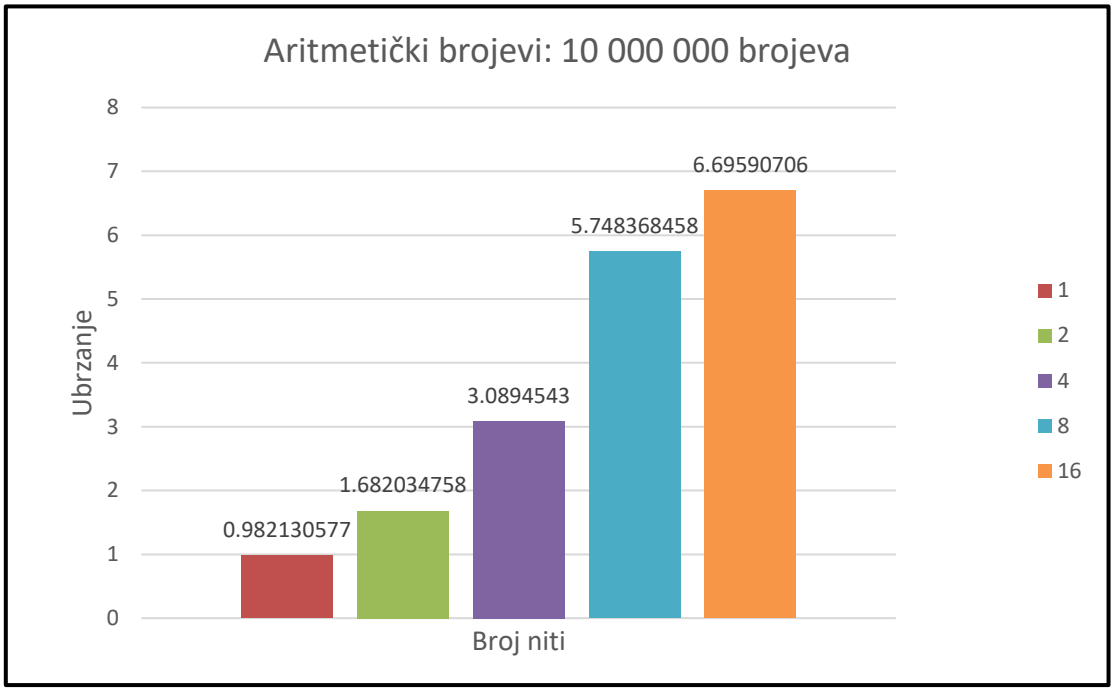
Slika 1. Aritmetički brojevi: 10 000 brojeva



Slika 2. Aritmetički brojevi: 100 000 brojeva



Slika 3. Aritmetički brojevi: 1 000 000 brojeva



Slika 4. Aritmetički brojevi: 10 000 000 brojeva

1.4.3. Diskusija dobijenih rezultata

Kao i kod rešenja pomoću worksharing direktiva, podela posla pomoću taskova takođe postiže dobro ubrzanje za različite brojeve niti i veličine problema. U trenutnoj implementaciji je raspodela posla takva da svaka nit dobije po jedan task kako bi granularnost posla bila što veća. Dodatno bi moglo da se testira ubrzanje ukoliko bi se povećao broj taskova za slučaj da postoji optimalnija raspodela posla.

2. PROBLEM 2 – HALTONQMC

2.1. Tekst problema

2.1.1. Zadatak 3

Paralelizovati program koji vrši generisanje elemenata Halton Quasi Monte Carlo (QMC) sekvence. Program se nalazi u datoteci halton.c u arhivi koja je priložena uz ovaj dokument. Prilikom paralelizacije nije dozvoljeno koristiti direktive za podelu posla (worksharing direktive), već je iteracije petlje koja se paralelizuje potrebno raspodeliti ručno. Obratiti pažnju na ispravno deklarisanje svih promenljivih prilikom paralelizacije. Program testirati sa parametrima koji su dati u run skripti.

2.1.2. Zadatak 4

Prethodni program paralelizovati korišćenjem direktiva za podelu posla (worksharing direktive). Program testirati sa parametrima koji su dati u run skripti.

2.2. Delovi koje treba paralelizovati

2.2.1. Diskusija

HaltonQMC je znatno lakši za paralelizaciju od problema izračunavanja aritmetičkih brojeva, pre svega zbog postojanja pravilnih pravougaonih petlji. Raspodela posla po iteracijama nije jednaka, međutim, u zavisnosti od ulaznih podataka, zahtevnije iteracije se mogu naći i na početku i na kraju petlje.

2.2.2. Zadatak 3 – Ručna paralelizacija

Postoje dve petlje koje je moguće paralelizovati. Prva inicijalizuje polja strukture za smeštanje podataka na nultu vrednost, dok druga vrši izračunavanje elemenata Haltonove sekvence. Paralelizacijom obe ove petlje može da se postigne ubrzanje, međutim, kako zbog režijskog troška dve odvojene omp for direktive, tako i zbog lokalnosti po podacima, doneta je odluka da se ove dve petlje sažmu u jednu. Svaka nit kada želi da sračunava vrednost određenog elementa sekvence inicijalizuje ga na početnu vrednost, a zatim odmah radi i izračunavanje. Testiranjem implementacije sa worksharing direktivama, uočeno je da najbolje rezultate daje ciklična raspodela, te je zato takva raspodela i ručno implementirana.

2.2.3. Zadatak 4 – Paralelizacija pomoću worksharing direktiva

Kao i kod prethodnog rešenja, i ovde su spojene petlje za inicijalizaciju i za računanje vrednosti elemenata. Eksperimentisanjem sa static raspodelom i veličinom blokova, uočeno je da se najbolje ubrzanje postiže kad je veličina bloka 1, odnosno, pri cikličnoj raspodeli.

2.3. Rezultati – Zadatak 3

U okviru ove sekcije su izloženi rezultati paralelizacije koda za izračunavanja elemenata Haltonove sekvence ručnom paralelizacijom.

2.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti.

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.006s
  user    0m0.005s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.042s
  user    0m0.038s
  sys     0m0.004s

HALTON_TEST:
```

```
HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647     0.00113823     0.00101176    0.000885292    0.000758821
  999:      0.00126279     0.00113651     0.00101023     0.00088395    0.000757671

  Normal end of execution.

  real    0m2.504s
  user    0m2.504s
  sys     0m0.000s
```

Listing1. HaltonQMC: Sekvencijalno izvršavanje

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.005s
  user    0m0.005s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
```

```
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.023s
  user    0m0.019s
  sys     0m0.004s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m1.974s
  user    0m1.974s
  sys     0m0.000s
```

Listing2. HaltonQMC: Zadatak 3 – Izvršavanje za 1 nit

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
```

```
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.005s
  user    0m0.007s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.021s
  user    0m0.039s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
```

```
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m1.149s
  user    0m2.297s
  sys     0m0.000s
```

Listing3. HaltonQMC: Zadatak 3 – Izvršavanje za 2 niti

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.004s
  user    0m0.007s
  sys     0m0.002s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906
```

```
  Normal end of execution.

  real    0m0.012s
  user    0m0.039s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m0.543s
  user    0m2.171s
  sys     0m0.000s
```

Listing4. HaltonQMC: Zadatak 3 – Izvršavanje za 4 niti

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.005s
```

```
  user    0m0.002s
  sys     0m0.012s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.010s
  user    0m0.051s
  sys     0m0.007s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m0.388s
  user    0m3.092s
  sys     0m0.004s
```

Listing5. HaltonQMC: Zadatak 3 – Izvršavanje za 8 niti

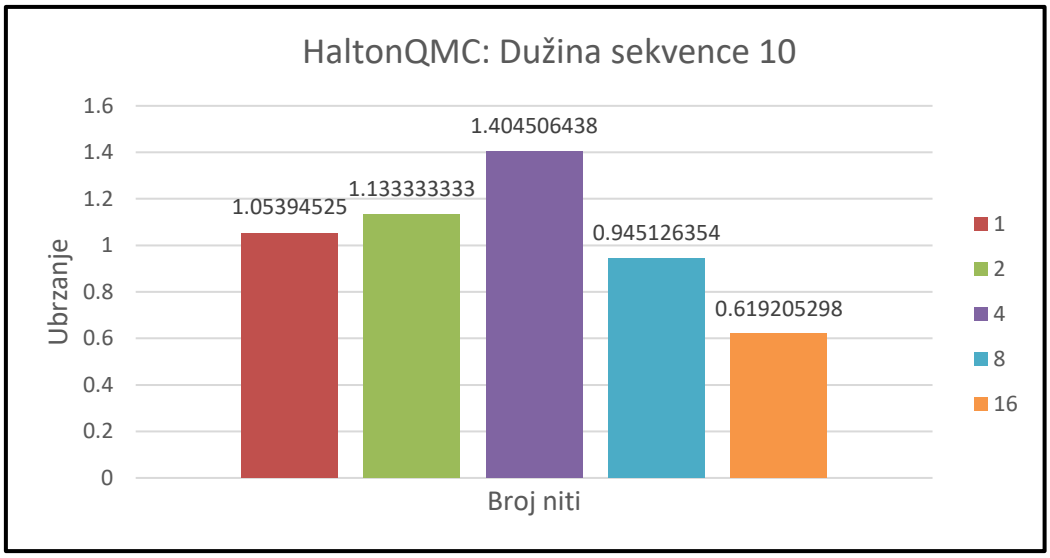
```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.005s
  user    0m0.000s
  sys     0m0.032s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.013s
  user    0m0.163s
  sys     0m0.005s

HALTON_TEST:

HALTON_SEQUENCE_TEST
```

```
HALTON_SEQUENCE returns the elements I1 through I2
of an M-dimensional Halton sequence.

R:
Col:          0             1             2             3             4
Row
  0:          0.3125          0.5625          0.0625          0.875           0.375
  ...
998:       0.0012647     0.00113823     0.00101176     0.000885292     0.000758821
999:       0.00126279     0.00113651     0.00101023      0.00088395     0.000757671

Normal end of execution.

real    0m0.464s
user    0m7.343s
sys     0m0.004s
```

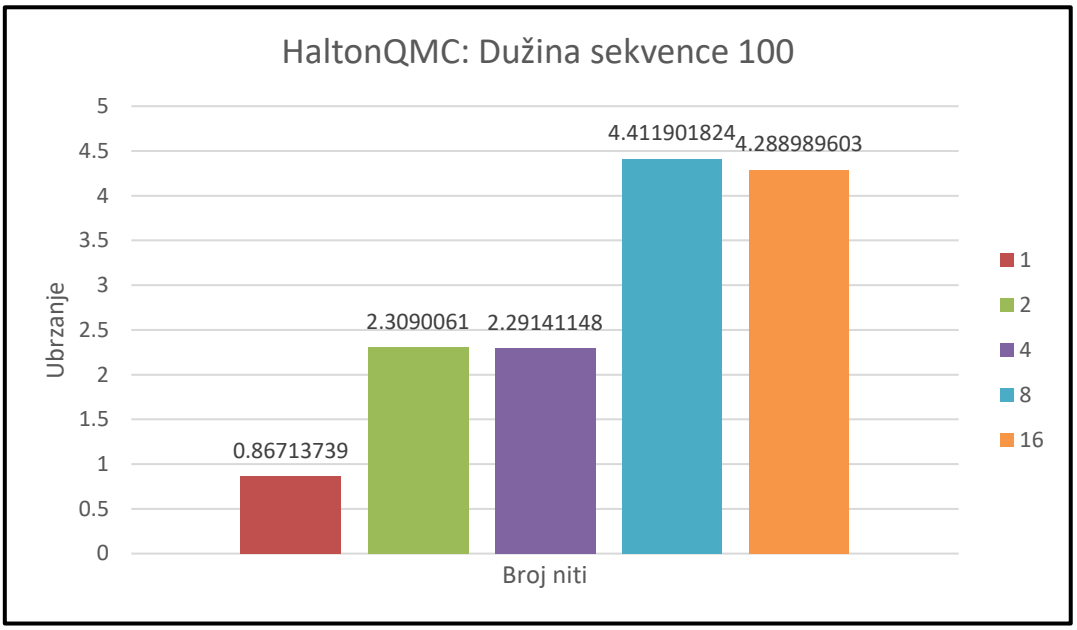
Listing6. HaltonQMC: Zadatak 3 – Izvršavanje za 16 niti

2.3.2. Grafici ubrzanja

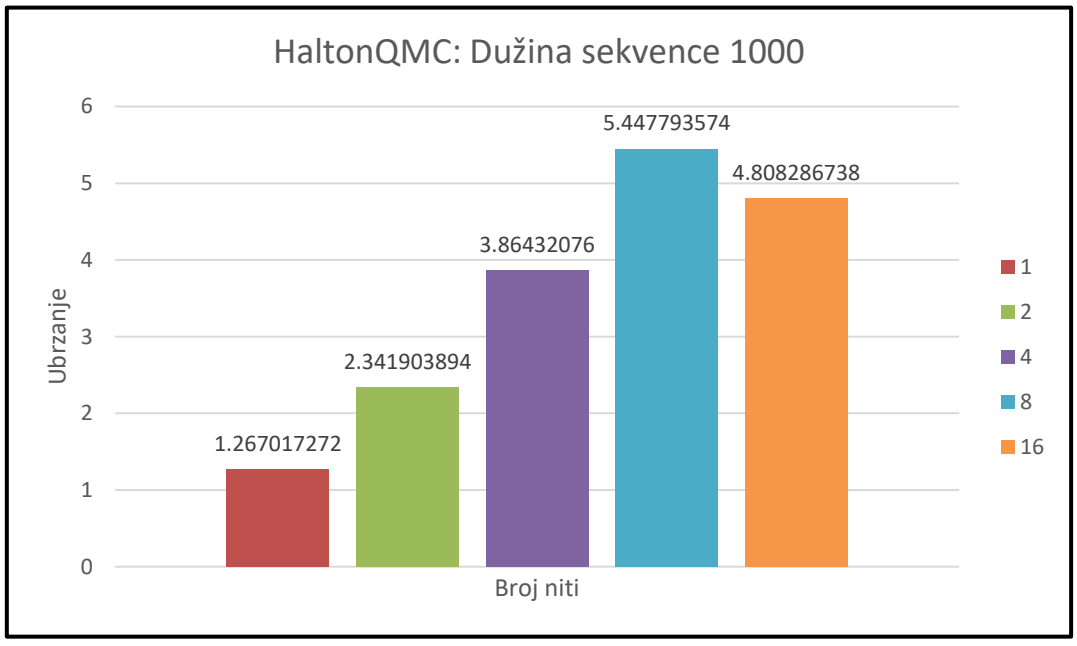
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



Slika 1. HaltonQMC: Dužina sekvence 10



Slika 2. HaltonQMC: Dužina sekvence 100



Slika 3. HaltonQMC: Dužina sekvence 1000

2.3.3. Diskusija dobijenih rezultata

Problem koji se javlja u dobijenim rezultatima jeste da se za neke brojeve niti dobija veće ubrzanje od broja niti što se kosi sa Amdalovim zakonom. Ovaj rezultat se dobija za pokretanje programa sa 1 ili 2 niti. S obzirom na sažimanje dveju petlji, moguće je da je donekle optimizovana sekvencijalna optimizacija, zbog lokalnosti po podacima. Takođe je verovatno i da sami podaci nisu najpreciziji zbog malih razlika u vremenima izvršavanja i nepredvidivim opterećenjima servera za izradu. Svakako, dobijeni podaci govore o tome da je postignuto značajno ubrzanje.

2.4. Rezultati – Zadatak 4

U okviru ove sekcije su izloženi rezultati paralelizacije koda za izračunavanja elemenata Haltonove sekvence ručnom paralelizacijom.

2.4.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti.

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.006s
  user    0m0.005s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
```

```
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.042s
  user    0m0.038s
  sys     0m0.004s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m2.504s
  user    0m2.504s
  sys     0m0.000s
```

Listing1. HaltonQMC: Sekvencijalno izvršavanje

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
```

```
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.004s
  user    0m0.004s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.023s
  user    0m0.023s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
```

```
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m1.614s
  user    0m1.614s
  sys     0m0.000s
```

Listing2. HaltonQMC: Zadatak 4 – Izvršavanje za 1 nit

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.004s
  user    0m0.005s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906
```

```
  Normal end of execution.

  real    0m0.017s
  user    0m0.032s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m1.008s
  user    0m2.011s
  sys     0m0.004s
```

Listing3. HaltonQMC: Zadatak 4 – Izvršavanje za 2 niti

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.004s
```

```
  user    0m0.007s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.012s
  user    0m0.039s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m0.544s
  user    0m2.173s
  sys     0m0.000s
```

Listing4. HaltonQMC: Zadatak 4 – Izvršavanje za 4 niti

```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.004s
  user    0m0.011s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.010s
  user    0m0.061s
  sys     0m0.000s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
```

```
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m0.384s
  user    0m3.066s
  sys     0m0.000s
```

Listing5. HaltonQMC: Zadatak 4 – Izvršavanje za 8 niti

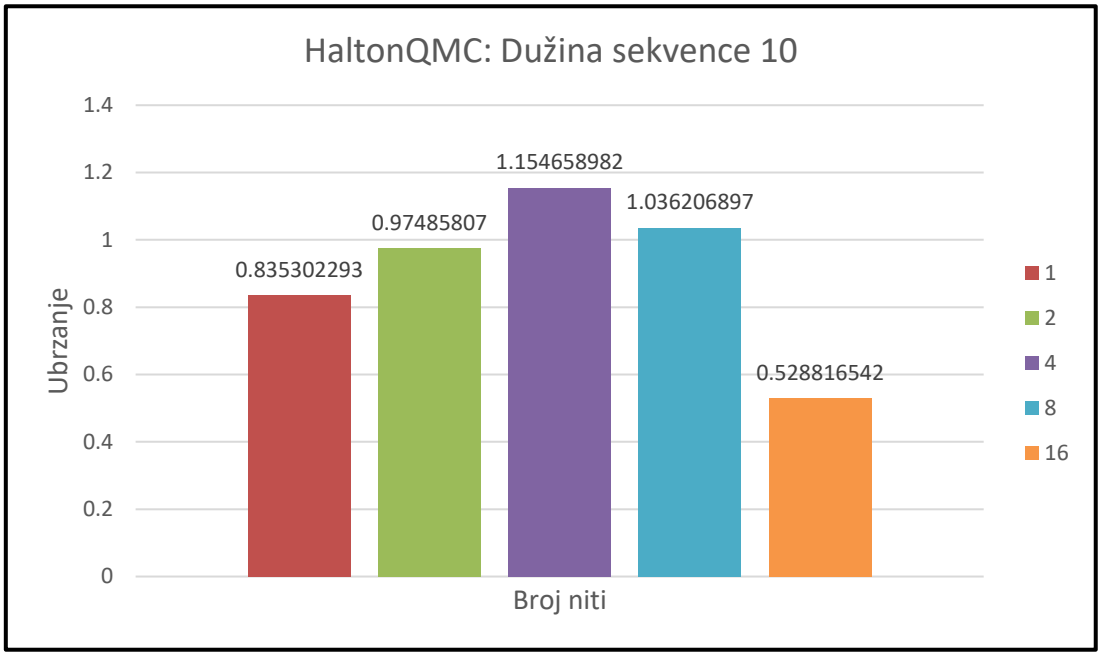
```
HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
    8:        0.434783        0.391304        0.347826        0.304348         0.26087
    9:        0.344828        0.310345        0.275862        0.241379        0.206897

  Normal end of execution.

  real    0m0.006s
  user    0m0.037s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
```

```
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
...
   98:       0.0191205       0.0172084       0.0152964       0.0133843       0.0114723
   99:       0.0184843       0.0166359       0.0147874        0.012939       0.0110906

  Normal end of execution.

  real    0m0.011s
  user    0m0.125s
  sys     0m0.001s

HALTON_TEST:

HALTON_SEQUENCE_TEST
  HALTON_SEQUENCE returns the elements I1 through I2
  of an M-dimensional Halton sequence.

  R:
  Col:          0             1             2             3             4
  Row
    0:          0.3125          0.5625          0.0625          0.875           0.375
    ...
  998:       0.0012647      0.00113823      0.00101176     0.000885292     0.000758821
  999:      0.00126279      0.00113651      0.00101023      0.00088395     0.000757671

  Normal end of execution.

  real    0m0.444s
  user    0m7.070s
  sys     0m0.000s
```

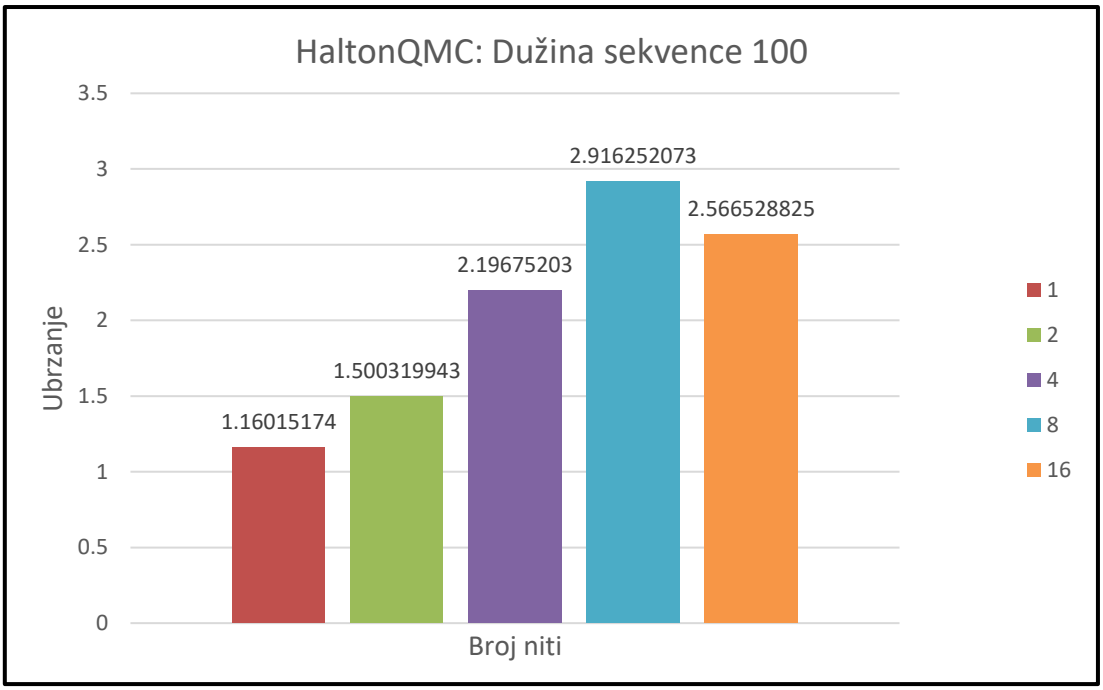
Listing6. HaltonQMC: Zadatak 4 – Izvršavanje za 16 niti

2.4.2. Grafici ubrzanja

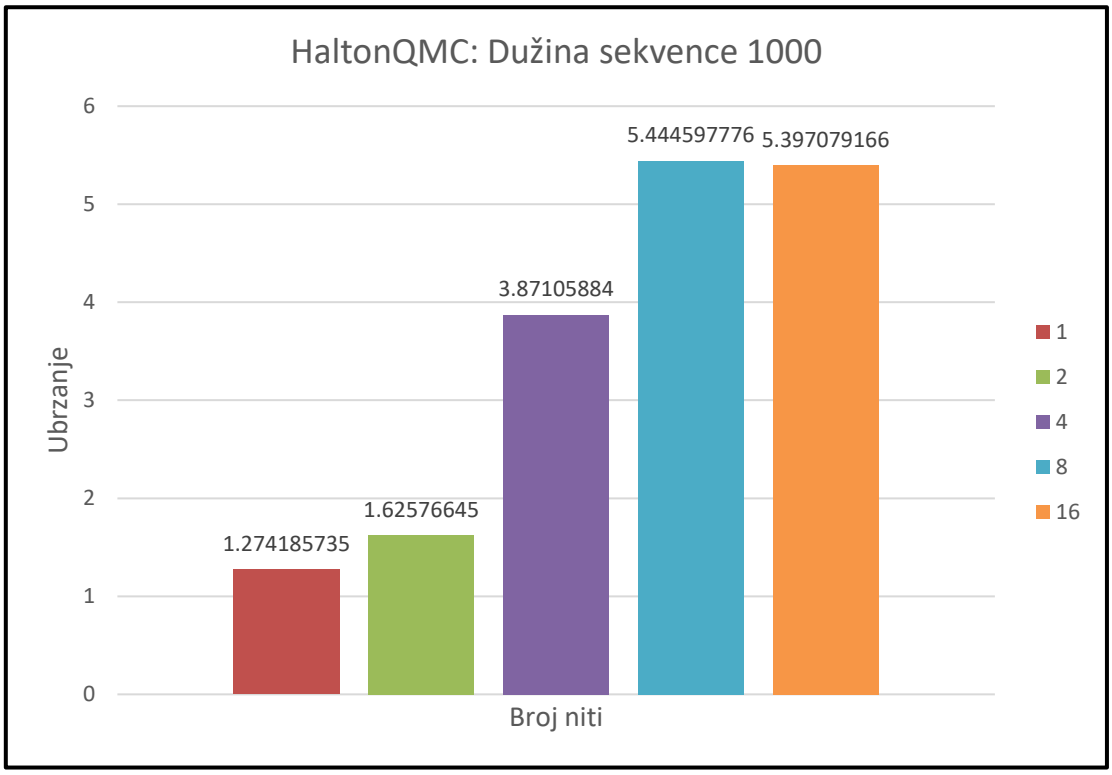
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



Slika 1. HaltonQMC: Dužina sekvence 10



Slika 2. HaltonQMC: Dužina sekvence 100



Slika 3. HaltonQMC: Dužina sekvence 1000

2.4.3. Diskusija dobijenih rezultata

Veći brojevi niti se bolje pokazuju za veće ulazne podatke. Za manje ulazne podatke nije postignuto značajno ubrzanje. Moguć razlog ovog problema je zbog režijskih troškova omp-a i raspodele posla među nitima pomoću omp for direktive.

3. PROBLEM 3 – N BODY PROBLEM

3.1. Tekst problema

3.1.1. Zadatak 5

Paralelizovati program koji se bavi problemom n tela (n-body problem). Sva tela imaju jediničnu masu, trokomponentni vektor položaja (x, y, z) i trokomponentni vektor brzine (vx, vy, vz). Simulaciju n tela se odvija u iteracijama, pri čemu se u svakoj iteraciji izračunava sila kojom sva tela deluju na sva ostala, a zatim se brzine i koordinate tela ažuriraju prema II Njutnovom zakonu. Brzine i položaji su slučajno generisani na početku simulacije. Zbog same prirode numeričke simulacije uveden je parametar SOFTENING, koji predstavlja korektivni faktor prilikom izračunavanja rastojanja između čestica (kako je gravitaciona sila obrnuto proporcionalna rastojanju između čestica, za nulta rastojanja i rastojanja bliska nuli, izračunata gravitaciona sila postaje izuzetno velika – teži beskonačnosti).

Program se nalazi u datoteci direktorijumu nbodymini u arhivi koja je priložena uz ovaj dokument. Program koji treba paralelizovati nalazi se u datoteci nbody.c. Pored samog izračunavanja, program čuva rezultate svake iteracije u zasebnim datotekama (za svako telo se čuvaju pozicije i brzine), dok kod show_nbody.py kreira gif same simulacije.

Ukoliko je potrebno međusobno isključenje prilikom paralelizacije programa, koristiti kritične sekcije ili atomske operacije. Skripta run pokreće simulaciju za različite parametre, i nakon toga, za određene simulacije poziva python kod koji kreira gifove.

3.2. Delovi koje treba paralelizovati

3.2.1. Diskusija

N Body Problem u svakoj iteraciji glavne petlje u main funkciji ažurira brzine tela na osnovu njihovog položaja u odnosu na druga tela. Zatim, položaji i brzine se sačuvaju u .csv fajl, nakon čega se ažuriraju položaji tela za narednu iteraciju. Zbog zavisnosti po podacima između iteracija koje je potrebno zabeležiti, nije moguće raspodeliti iteracije simulacije po nitima. Dodatan problem su upisi u fajlove u svakoj iteraciji simulacije, koje nije moguće paralelizovati. Sa druge strane, moguće je raspodeliti ažuriranje brzine i položaja tela između niti, tako da svaka nit dobije svoju grupu tela sa kojom radi.

3.2.2. Zadatak 5

Paralelizam je u ovom zadatku postignut raspodelom ažuriranja brzine i položaja tela između niti, tako da svaka nit dobije svoju grupu tela sa kojom radi. Kako bi se izbeglo pravljenje dva paralelna regiona u istoj iteraciji i pravljenje single bloka, inicijalno ažuriranje brzina i upis u fajl za početnu iteraciju smešteni su pre petlje za iteracije simulacije. Ta petlja sada polazi od prve, a ne od nulte iteracije. Ažuriranje položaja tela i sračunavanje brzine sad mogu da se stave pod isti parallel blok, dok se upis u odgovarajući fajl odvija na kraju iteracije petlje.

3.3. Rezultati – Zadatak 5

U okviru ove sekcije su izloženi rezultati paralelizacije koda za izračunavanja problema n tela.

3.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti.

```
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m28.404s
user    0m28.064s
sys     0m0.240s
```

Listing 1. NBodyMini: Zadatak 5 – Sekvencijalno izvršavanje

```
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m26.549s
user    0m26.208s
sys     0m0.258s
```

Listing 2. NBodyMini: Zadatak 5 – Izvršavanje za 1 nit

C

```
HALTON_SEQUENCE_TEST
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m15.804s
user    0m31.272s
sys     0m0.274s
```

Listing 3. NBodyMini: Zadatak 5 – Izvršavanje za 2 niti

```
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m10.205s
user    0m39.861s
sys     0m0.227s
```

Listing 4. NBodyMini: Zadatak 5 – Izvršavanje za 4 niti

```
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m9.335s
user    1m13.026s
sys     0m0.357s
```

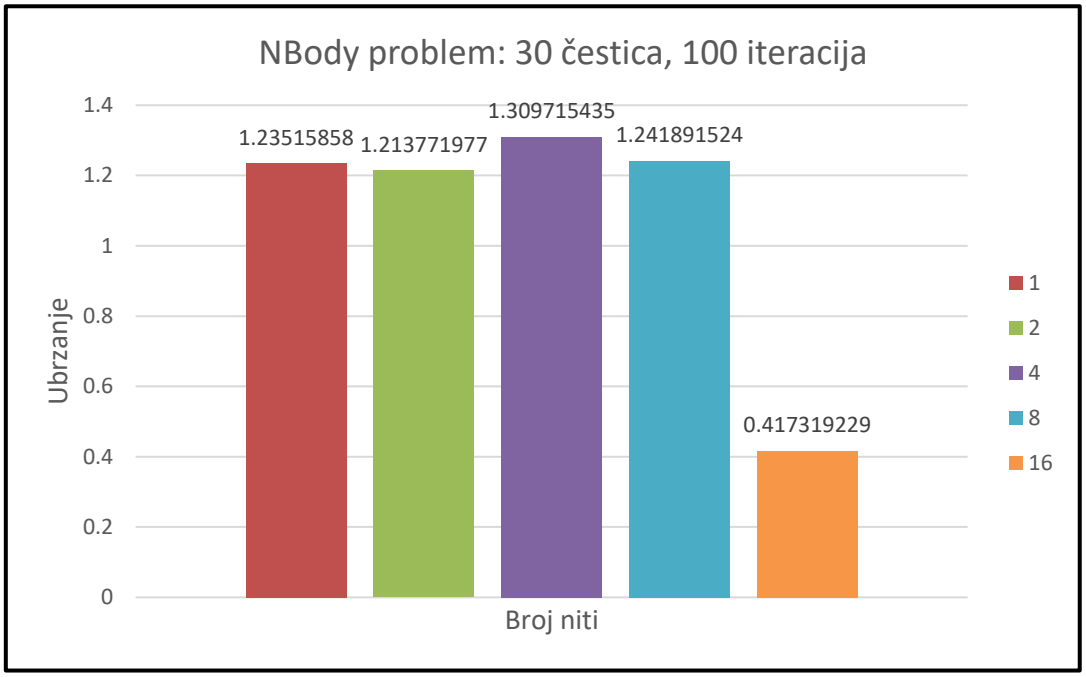
Listing 5. NBodyMini: Zadatak 5 – Izvršavanje za 8 niti

```
Running ./nbody with 30 particles, 100 iterations, and saving to simulation_1 folder
Running ./nbody with 30 particles, 1000 iterations, and saving to simulation_2 folder
Running ./nbody with 3000 particles, 100 iterations, and saving to simulation_3 folder
Running ./nbody with 3000 particles, 1000 iterations, and saving to simulation_4 folder

real    0m14.891s
user    3m41.395s
sys     0m0.592s
```

Listing 6. NBodyMini: Zadatak 5 – Izvršavanje za 16 niti

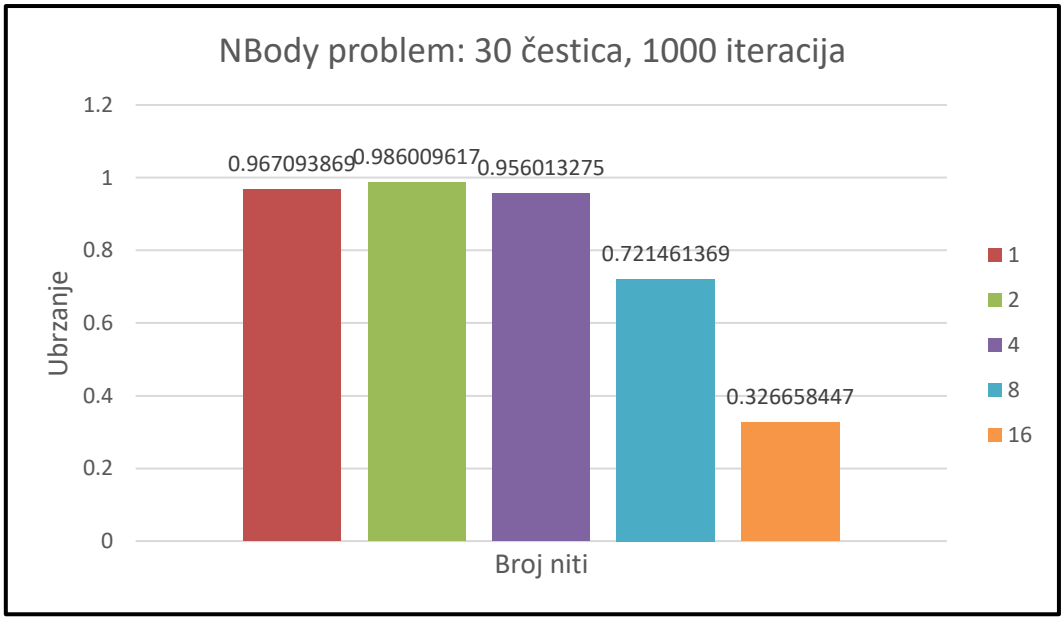
3.3.2. Grafici ubrzanja

U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju

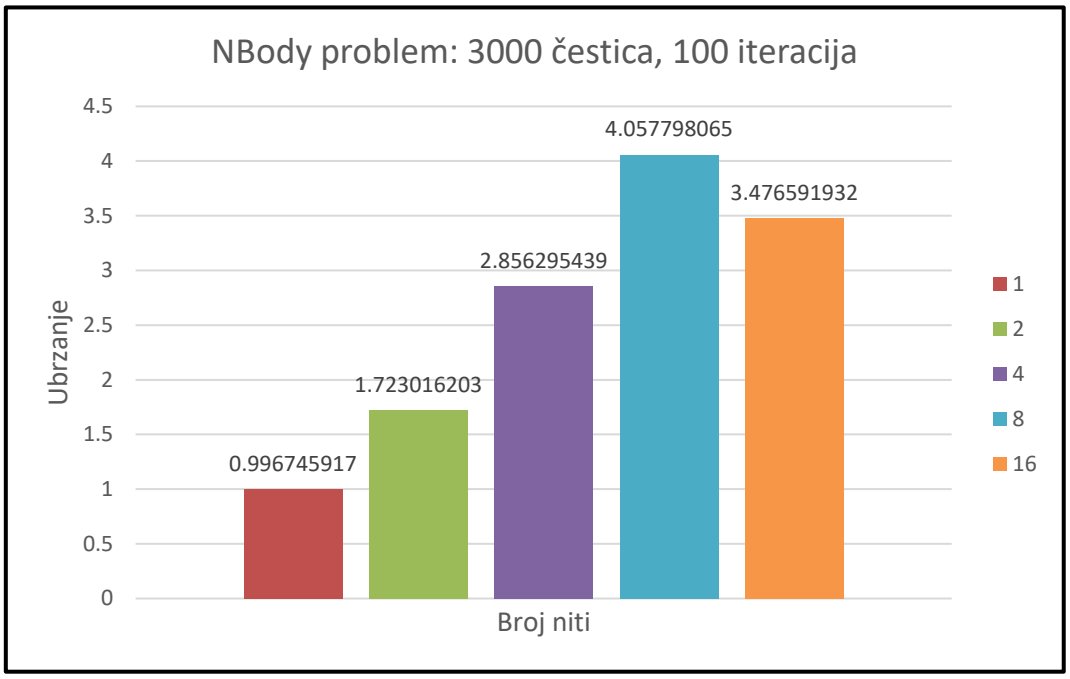


.

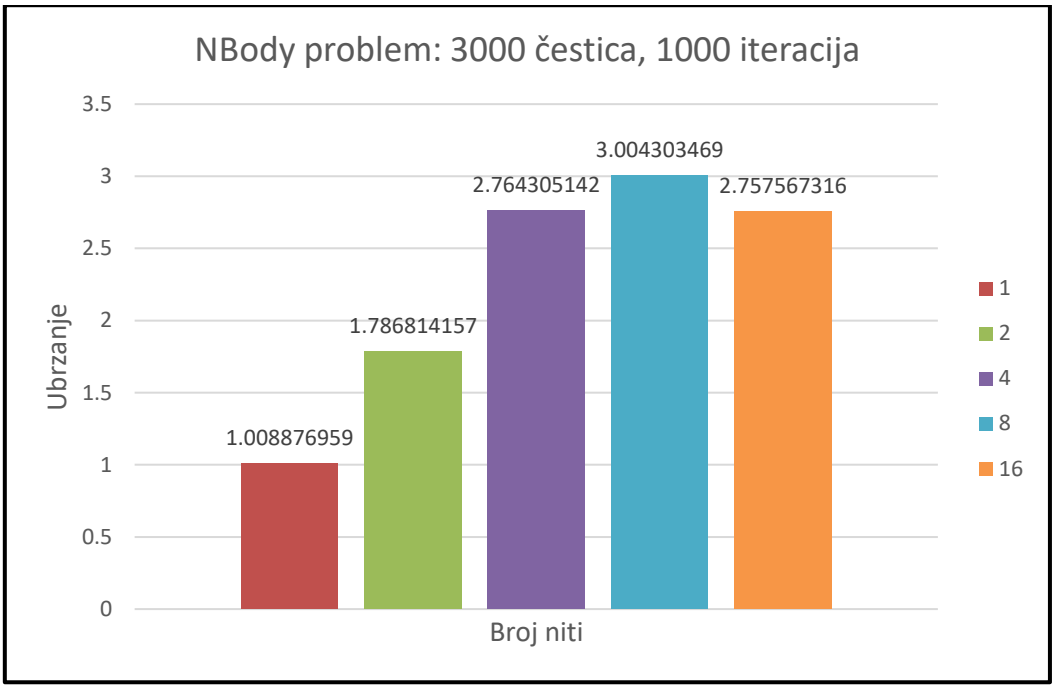
Slika 1. Nbody problem: 30 čestica, 100 iteracija



Slika 2. Nbody problem: 30 čestica, 1000 iteracija



Slika 3. Nbody problem: 3000 čestica, 100 iteracija



Slika 4. Nbody problem: 3000 čestica, 1000 iteracija

3.3.3. Diskusija dobijenih rezultata

S obzirom na raspodelu posla po telima, a ne po iteracijama simulacije, sa priloženih grafika se može izvući par zaključaka. Ubrzanje je veće kada ima više nebeskih tela, a manje iteracija simulacije. Sa porastom broja iteracija, raste broj ulazaka u paralelni region, što sa sobom povlači usporenje zbog režijskih troškova. Takođe raste i sekvencijalni deo koda i čekanje zbog upisivanja u fajlove. Porastom broja nebeskih tela, a smanjenjem iteracija, veći deo posla biva odrađen u paraleli.