UNIVERZITET U BEOGRADU - ELEKTROTEHNIČKI FAKULTET

MULTIPROCESORKI SISTEMI (13S114MUPS, 13E114MUPS)

DOMAĆI ZADATAK 1 – OPENMP

Izveštaj o urađenom domaćem zadatku

Predmetni asistent:

doc. dr Marko Mišić

Studenti:

Ema Pajić 2016/0017

Nikola Aleksić 2016/0022

Beograd, mart 2020.

SADRŽAJ

1. PROBLEM 1 – JULIA SET

1.1. Tekst problema

Paralelizovati program koji formira sliku tačaka koje pripadaju Julia skupu tačaka (https://en.wikipedia.org/wiki/Julia_set). Neka se posmatra skup tačaka (x, y) u na pravougaonom domenu x, y ∈ [-1,5, 1.5] i neka važi z = x+yi. Julia skup je skup tačaka za koji iteracija z = z2 + c ne divergira za određene zadate početne uslove. U zadatom programu početni uslov odgovara c=-0.8+0.156i. Ukoliko u bilo kom trenutku važi 1000 < |z|, smatra se da tačka z ne pripada Julia skupu. Program formira sliku u Targa (.tga) formatu koja se može otvoriti u nekom od namenskih pregledača slika. Program se nalazi u datoteci julia.c u arhivi koja je priložena uz ovaj dokument, dok se primeri izlaznih datoteka nalaze u direktorijumu output. 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 datoteci run. [1, N]

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

U funkciji julia_set uočene su 2 ugnježđene for petlje koje je moguće paralelizovati. Postoji i for u funkciji julia, ali njega ne vredi paralelizovati jer bi granularnost bila prevelika.

1.2.2. Način paralelizacije

Kako u toku rada ovog zadatka nije dozvoljeno korišćenje worksharing direktiva, paralelizujemo ručno određivanjem promenljivih chunk, start i end koje će svakoj niti na osnovu njene ID vrednosti dodeliti neki deo for petlje. Pošto treba paziti na balansiranost opterećenja niti, a tačke ne konvergiraju istom brzinom (neke i divergiraju), ako nitima dodelimo samo geometrijski bliske tačke može da se desi da je neka nit mnogo opterećenija od neke druge. Iz tog razloga pravimo po 10 razdvojenih blokova za svaku nit (koji idu kružno) i tome služe promenljive offset i cyclicChunk.

1.3. Rezultati

Vremena sekvencijalnog i paralelnog izvrsavanja u tabeli su srednje vrednosti 3 pokretanja programa sa zadatim parametrima i navedeni su u formatu (vreme sekvencijalnog izvršavanja) / (vreme

paralelnog izvršavanja) = ubrzanje. Za racunanje srednjeg ubrzanja nisu racunata pokretanja za prva tri ulaza zato sto je racunanjem na 2 decimale izgubljena preciznost. Pokretanjem sa 3 decimale smo videli da takodje daju otprilike isto ubrzanje kao za vece ulaze.

broj niti / ulazi (h, w, cnt)	1	2	4	8
500 500 200	0.043/0.04 = 1.07	0.05/0.02 = 2.5	0.046/0.01 = 4.6	0.046/0.01 = 4.6
500 500 500	0.05/0.05 = 1	0,05/0.02 = 2.5	0.05/0.01 = 5	0.05/0.01 = 5
500 500 1000	0.05/0.05 = 1	0.05/0.02 = 2.5	0.05/0.01 = 5	0.05/0.01 = 5
1000 1000 200	0.15/0.15 = 1	0.15/0.08 = 1.875	0.15/0.04 = 3.75	0.15/0.03 = 5
1000 1000 500	0.19/0.19 = 1	0.19/0.093 = 2.04	0.19/0.05 = 3.8	0.193/0.03 = 6.43
1000 1000 1000	0.20/0.19 = 1.05	0.193/0.10 = 1.93	0.19/0.05 = 3.8	0.19/0.03 = 6.33
2000 1000 200	0.31/0.303 = 1.02	0.303/0.15 = 2.02	0.30/0.08 = 3.75	0.30/0.05 = 6
2000 1000 500	0.38/0.373 = 1.02	0.38/0.19 = 2	0.38/0.10 = 3.8	0.383/0.073 = 5.2
2000 1000 1000	0.393/0.39 = 1.01	0.39/0.193 = 2.02	0.39/0.10 = 3.9	0.39/0.07 = 5.57
Srednje ubrzanje bez prva tri ulaza	**1.02**	**1.98**	**3.8**	**5.75**

1.3.1. Logovi izvršavanja

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

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
```

```
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.15
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.30
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.37
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'
```

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.39
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 1. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 1 niti Julia

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.08
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
```

```
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.15
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.40
Elapsed time, parallel: 0.20
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 2. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 2 niti Julia

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED
```

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.16
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.08
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 3. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 4 niti Julia

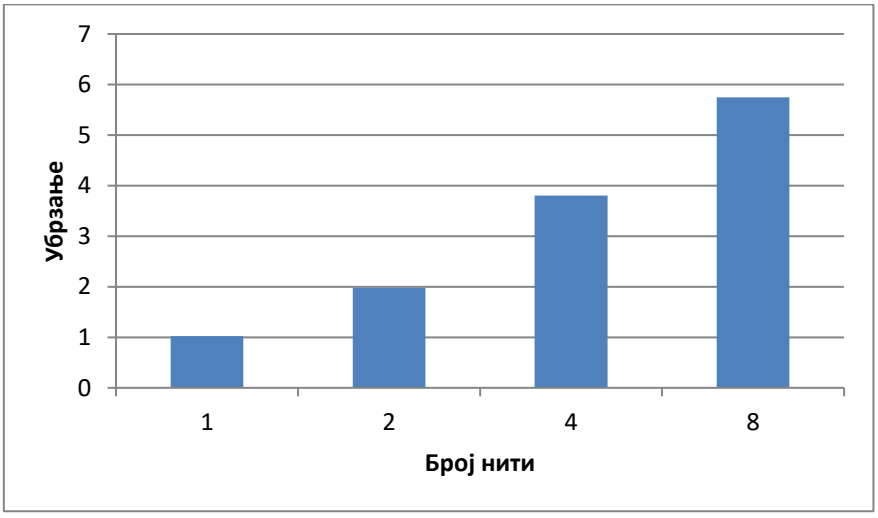
```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.05
Test PASSED
```

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.07
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.07
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 4. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 8 niti Julia

1.3.2. Grafici ubrzanja

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



Slika 1. Grafik zavisnosti ubrzanja algoritma bez worksharing direktiva od broja niti

1.3.3. Diskusija dobijenih rezultata

Paralelizacija je ubrzala izvršavanje za 2, 4 i 8 niti, a pri radu sa 1 niti nije usporila. Problem je embarrassingly parallel i svaka nit dobija da računa pripadnost Julia setu za deo tačaka, jedino je potrebno paziti na opterećenje po nitima. Pošto tačke ne konvergiraju istom brzinom (a neke i divergiraju), ako nitima dodelimo samo geometrijski bliske tačke može da se desi da je neka nit

mnogo opterećenija od neke druge. Iz tog razloga pravimo po 10 razdvojenih blokova za svaku nit (koji idu kružno). Najveće ubrzanje se dostiglo sa 8 niti gde primećujemo da je za veće dimenzije problema ubrzanje značajno u odnosu na izvršavanje sa 4 niti. Što se tiče manjih dimenzija problema (500x500), povećavanje broja niti sa 4 na 8 ne doprinosi daljem ubrzanju.

2. PROBLEM 2 – JULIA SET

2.1. Tekst problema

Prethodni program paralelizovati korišćenjem direktiva za podelu posla (worksharing direktive). Obratiti pažnju na raspodelu opterećenja po nitima i testirati program za različite načine raspoređivanja posla. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

2.2. Delovi koje treba paralelizovati

2.2.1. Diskusija

U funkciji julia_set uočene su 2 ugnježdene for petlje koje je moguće paralelizovati. Postoji i for u funkciji julia, ali njega ne vredi paralelizovati jer bi granularnost bila prevelika.

2.2.2. Način paralelizacije

U tekstu zadatka rečeno je da se paralelizacija izvrši korišćenjem worksharing direktive, pa je korišćen omp parallel for. Pošto imamo 2 ugnježdene for petlje, iskorišćeno je collapse(2). Probali smo različite načine shedulovanja i ispostavilo se da najbolje radi ako stavimo schedule(static, 10).

2.3. Rezultati

Vremena sekvencijalnog i paralelnog izvrsavanja u tabeli su srednje vrednosti 3 pokretanja programa sa zadatim parametrima i navedeni su u formatu (vreme sekvencijalnog izvršavanja) / (vreme paralelnog izvršavanja) = ubrzanje. Za racunanje srednjeg ubrzanja je uradjeno kao u prvom zadatku.

broj niti / ulazi (h, w, cnt)	1	2	4	8
500 500 200	0.043/0.04 = 1.07	0.047/0.02 = 2.35	0.043/0.01 = 4.3	0.043/0.01 = 4.3
500 500 500	0.05/0.05 = 1	0,05/0.02 = 2.5	0.05/0.01 = 5	0.05/0.01 = 5
500 500 1000	0.05/0.05 = 1	0.05/0.02 = 2.5	0.05/0.01 = 5	0.05/0.01 = 5
1000 1000 200	0.15/0.15 = 1	0.15/0.08 = 1.875	0.15/0.04 = 3.75	0.15/0.03 = 5
1000 1000 500	0.19/0.19 = 1	0.19/0.097 = 1.96	0.19/0.05 = 3.8	0.19/0.03 = 6.33
1000 1000 1000	0.197/0.19 = 1.03	0.19/0.10 = 1.9	0.19/0.05 = 3.8	0.19/0.03 = 6.33
2000 1000 200	0.307/0.30 = 1.02	0.303/0.15 = 2.02	0.303/0.08 = 3.79	0.30/0.05 = 6
2000 1000 500	0.383/0.38 = 1.01	0.38/0.19 = 2	0.38/0.093 = 4.08	0.38/0.06 = 6.33
2000 1000 1000	0.39/0.39 = 1	0.39/0.20 = 1.95	0.39/0.10 = 3.9	0.39/0.067 = 5.82
Srednje ubrzanje bez prva tri ulaza	**1.01**	**1.95**	**3.85**	**5.97**

2.3.1. Logovi izvršavanja

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

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'
```

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.16
Elapsed time, parallel: 0.15
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.31
Elapsed time, parallel: 0.30
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.38
Test PASSED
```

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.39
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 5. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 1 niti Julia2

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.16
Elapsed time, parallel: 0.08
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.09
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'
```

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.31
Elapsed time, parallel: 0.16
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.37
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.20
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 6. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 2 niti Julia2

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
pe160017d@rtidev5:~/pe160017d/pe160017d$ OMP_NUM_THREADS=4 ./run_dz2

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
```

```
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.06
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.06
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.37
Elapsed time, parallel: 0.11
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.44
```

```
Elapsed time, parallel: 0.11
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 7. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 4 niti Julia2

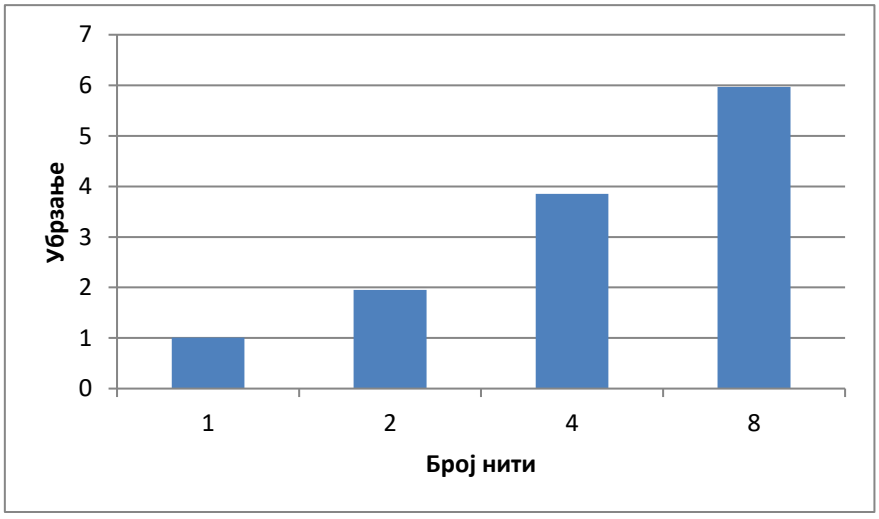
```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.03
Test PASSED
```

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.31
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.06
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.07
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 8. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 8 niti Julia2

2.3.2. Grafici ubrzanja

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



Slika 2. Grafik zavisnosti ubrzanja algoritma pomocu for worksharing direktive od broja niti

2.3.3. Diskusija dobijenih rezultata

Ovaj zadatak je isti kao prethodni, samo što je zadatak bila paralelizacija korišćenjem worksharing direktive. Iz tog razloga važe slični zaključci kao kao i u prethodnom zadatku. Paralelizacija je ubrzala izvršavanje za 2, 4 i 8 niti, a pri radu sa 1 niti nije usporila. Problem je embarrassingly parallel i svaka nit dobija da računa pripadnost Julia setu za deo tačaka, jedino je potrebno paziti na opterećenje po nitima. Pošto tačke ne konvergiraju istom brzinom (a neke i divergiraju), ako nitima dodelimo samo geometrijski bliske tačke može da se desi da je neka nit mnogo opterećenija od neke druge. Iz tog razloga je nije dobar samo schedule(static). Probali smo i guided i dynamic jer su delovali kao dobar izbor, ali ipak se najbolje pokazao schedule(static, 10). Najveće ubrzanje se dostiglo sa 8 niti gde primećujemo da je za veće dimenzije problema ubrzanje značajno u odnosu na izvršavanje sa 4 niti. Što se tiče manjih dimenzija problema (500x500), povećavanje broja niti sa 4 na 8 ne doprinosi daljem ubrzanju.

3. PROBLEM 3 – JULIA SET

U okviru ovog poglavlja je dat kratak izveštaj u vezi rešenja zadatog problema 1.

3.1. Tekst problema

Rešiti prethodni problem korišćenjem koncepta poslova (tasks). Obratiti pažnju na eventualnu potrebu za sinhronizacijom i testirati program za različite granularnosti poslova. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

3.2. Delovi koje treba paralelizovati

3.2.1. Diskusija

U funkciji julia_set uočene su 2 ugnježdene for petlje koje je moguće paralelizovati. Postoji i for u funkciji julia, ali njega ne vredi paralelizovati jer bi granularnost bila prevelika.

3.2.2. Način paralelizacije

U tekstu zadatka je rečeno je da se paralelizacija izvrši korišćenjem koncepta poslova (tasks). Pošto imamo 2 ugnježdene for petlje, probali smo 2 varijante paralelizacije. U jednoj je kao task stavljen ceo unutrašnji for, a u drugoj se kao task računao samo poziv funkciji julia (što bi u suštini bilo kao da je računanje julia vrednosti svake tačke poseban task). U implementaciji gde se kao task računa poziv funkciji julia, program je radio sporije nego sekvencijalno, što znači da je granularnost bila prevelika. Ako se ceo unutrašnji for računao kao task, program je radio dobro i značajno je bio ubrzan.

3.3. Rezultati

Vremena sekvencijalnog i paralelnog izvrsavanja u tabeli su srednje vrednosti 3 pokretanja programa sa zadatim parametrima i navedeni su u formatu (vreme sekvencijalnog izvršavanja) / (vreme paralelnog izvršavanja) = ubrzanje. Za racunanje srednjeg ubrzanja je uradjeno kao u prvom zadatku.

broj niti / ulazi (h, w, cnt)	1	2	4	8
500 500 200	0.047/0.043 = 1.1	0.05/0.02 = 2.5	0.05/0.01 = 5	0.043/0.01 = 4.3
500 500 500	0.05/0.05 = 1	0,05/0.02 = 2.5	0.053/0.01 = 5.3	0.05/0.01 = 5
500 500 1000	0.05/0.05 = 1	0.05/0.02 = 2.5	0.05/0.01 = 5	0.05/0.01 = 5
1000 1000 200	0.15/0.15 = 1	0.15/0.08 = 1.875	0.15/0.04 = 3.75	0.15/0.03 = 5
1000 1000 500	0.19/0.19 = 1	0.19/0.09 = 2.11	0.19/0.05 = 3.8	0.19/0.03 = 6.33
1000 1000 1000	0.20/0.19 = 1.05	0.193/0.10 = 1.93	0.193/0.05 = 3.86	0.193/0.03 = 6.43
2000 1000 200	0.31/0.30 = 1.03	0.307/0.15 = 2.05	0.30/0.08 = 3.75	0.303/0.05 = 6.06
2000 1000 500	0.38/0.373 = 1.02	0.38/0.19 = 2	0.38/0.10 = 3.8	0.38/0.06 = 6.33
2000 1000 1000	0.397/0.397 = 1	0.39/0.19 = 2.05	0.39/0.10 = 3.9	0.39/0.07 = 5.57
Srednje ubrzanje bez prva tri ulaza	**1.01**	**2.00**	**3.81**	**5.95**

3.3.1. Logovi izvršavanja

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

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'
```

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.15
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.20
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.30
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.37
Elapsed time, parallel: 0.37
Test PASSED
```

```
TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.39
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 9. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 1 niti Julia3

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.02
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.08
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.09
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'
```

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.15
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.19
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 10. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 2 niti Julia3

```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.04
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
```

```
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.04
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.08
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.39
Elapsed time, parallel: 0.10
Test PASSED

TGA_WRITE:
```

```
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 11. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 4 niti Julia3

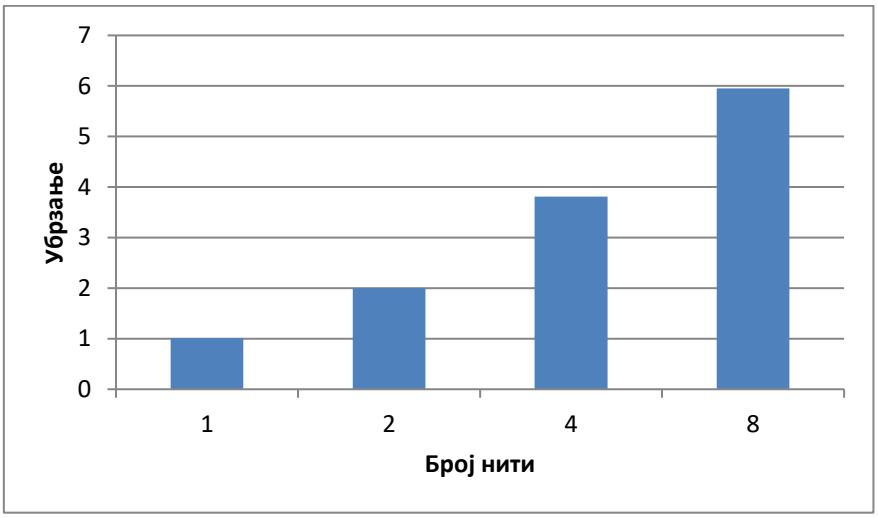
```
JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.05
Elapsed time, parallel: 0.01
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_500_500_1000.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.15
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.19
Elapsed time, parallel: 0.03
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_1000_1000_1000.tga'

JULIA Set
```

```
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.30
Elapsed time, parallel: 0.05
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_200.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.38
Elapsed time, parallel: 0.07
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_500.tga'

JULIA Set
  Plot a version of the Julia set for Z(k+1)=Z(k)^2-0.8+0.156i
Elapsed time, sequential: 0.40
Elapsed time, parallel: 0.07
Test PASSED

TGA_WRITE:
  Graphics data saved as 'output/dz_julia_2000_1000_1000.tga'
```

Listing 12. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 8 niti Julia3

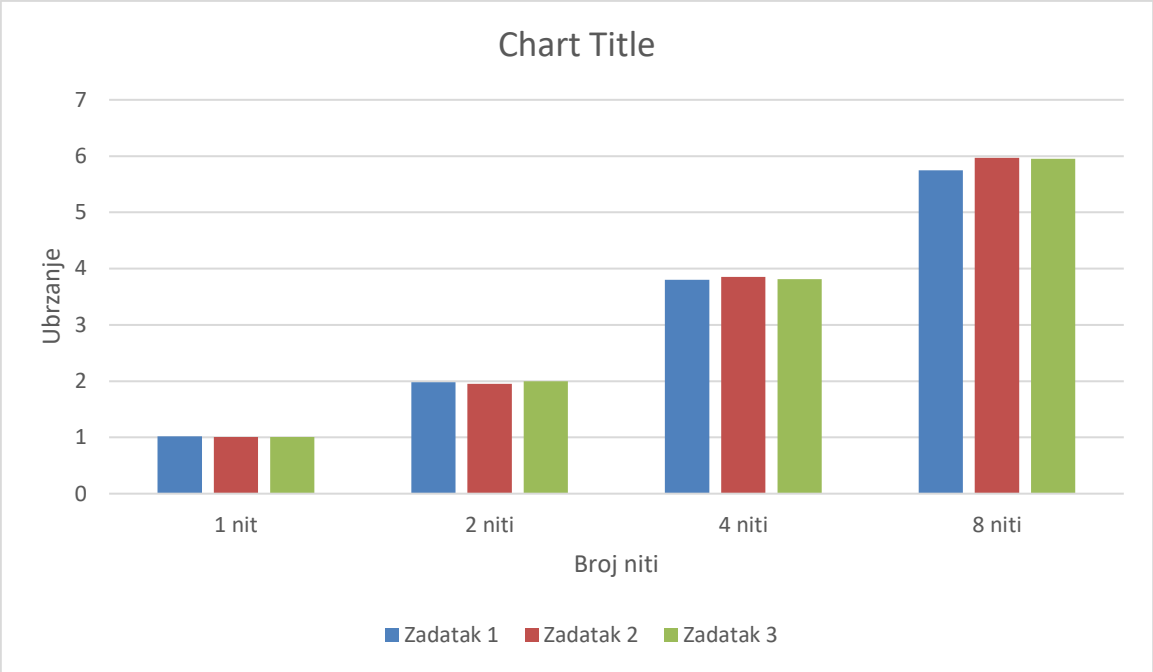
3.3.2. Grafici ubrzanja

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



Slika 3. Grafik zavisnosti ubrzanja algoritma pomocu taskova od broja niti

Takođe, ovde ćemo dati poređenje ubrzanja za 3 različite implementacije iz prva tri zadatka.



Slika 4. Grafik poređenja ubrzanja implementacija iz prva tri zadatka za različit broj niti

3.3.3. Diskusija dobijenih rezultata

Ovaj zadatak je isti kao prethodna dva, samo što je zadatak bila paralelizacija korišćenjem koncepta poslova. Iz tog razloga važe slični zaključci kao kao i u prethodnim zadacima. Paralelizacija je ubrzala izvršavanje za 2, 4 i 8 niti, a pri radu sa 1 niti nije usporila. Problem je embarrassingly parallel i svaka nit dobija da računa pripadnost Julia setu za deo tačaka. Kako ovde problem rešavamo konceptom poslova, svaki task će biti upakovan za kasnije izvršavanje o predata thread pool-u i time ćemo izbeći problem balansiranosti opterećenja. Jedino ostaje odabir obima posla koji ćemo staviti u task i ispostavilo se će granularnost problema biti prevelika ako računanje za svaku tačku pojedinačno računamo kao task, a da je dobro ukoliko ceo red piksela računamo u task. Najveće ubrzanje se dostiglo sa 8 niti gde primećujemo da je za veće dimenzije problema ubrzanje značajno u odnosu na izvršavanje sa 4 niti. Što se tiče manjih dimenzija problema (500x500), povećavanje broja niti sa 4 na 8 ne doprinosi daljem ubrzanju.

4. PROBLEM 4 – IZOŠTRAVANJE PGM SLIKA

4.1. Tekst problema

Paralelizovati program koji izoštrava zadatu sliku u Portable Graymap Format (PGM) formatu. PGM format se može otvoriti u nekom od namenskih pregledača slika ili online na adresi http://paulcuth.me.uk/netpbm-viewer/. Program se nalazi u direktorijumu sharpen u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih su od interesa datoteke sharpen.c, dosharpen.c i filter.c. Obratiti pažnju na raspodelu opterećenja po nitima i testirati program za različite načine raspoređivanja posla. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

4.2. Delovi koje treba paralelizovati

4.2.1. Diskusija

Paralelizacija je moguća u dosharpen funkciji. Glavni zadatak za paralelizaciju je računanje matrice konvolucije. Uočavmo da je takođe moguće paralelizovati još neke for petlje koje postavljaju vrednosti matricama fuzzy, sharp, fuzzyPadded, sharp i sharpCropped. U poglavlju diskusija dobijenih rezultata biće dalje diskutovano oko isplativosti tih dodatnih paralelizacija.

4.2.2. Način paralelizacije

Što se tiče paralelizacije računanja konvolucije, imamo 2 ugnježdene for petlje koje treba paralelizovati. Pokušali smo sa 2 pristupa, korišćenjem koncepta poslova (task) i worksharing direktive omp parallel for collapse (2). Oba načina su dala slične rezultate, ali nam se činilo da je ipak malo bolje bilo pomoću taskova pa je u implementaciji koju predajemo paralelizacija korišćenjem koncepta poslova. Takođe smo uočili da postoje još neke ugnježdene for petlje koje je moguće paralelizovati. One su paralelizovane korišćenjem worksharing direktive omp parallel for collapse(2), međutim njihova paralelizacija nije pravila veliku razliku u vremenu izvršavanja programa.

4.3. Rezultati

Vremena sekvencijalnog i paralelnog izvrsavanja u tabeli su srednje vrednosti 3 pokretanja programa sa zadatim parametrima i navedeni su u formatu (vreme sekvencijalnog izvršavanja) / (vreme paralelnog izvršavanja) = ubrzanje.

broj niti / slika i dimenzija	1	2	4	8
balloons_noisy 640x480	3.153/3.193 = 0.987	3.153/1.633 = 1.9308	3.157/0.867 = 3.641	3.267/0.657 = 4.97
bone_scint 1129x1865	21.29/21.357 = 0.997	21.28/10.91 = 1.9505	21.547/5.69 = 3.787	21.323/4.09 = 5.213
fuzzy 564x770	4.477/4.52 = 0.9905	4.41/2.30 = 1.9174	4.517/1.21 = 3.733	4.44/0.87 = 5.1034
lena512 512x512	2.75/2.697 = 1.019	2.68/1.413 = 1.8967	2.713/0.737 = 3.681	2.72/0.52 = 5.2308
man 1024x1024	10.63/10.67 = 0.996	10.78/5.573 = 1.934	10.637/2.893 = 3.6768	10.657/2.07 = 5.148
Rainier_blur 1920x1080	21.03/21.19 = 0.992	21.123/10.87 = 1.943	20.957/5.577 = 3.755	20.983/3.983 = 5.268
Srednje ubrzanje bez prva tri ulaza	**0.99**	**1.93**	**3.71**	**5.16**

4.3.1. Logovi izvršavanja

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

```
Input file is: data/balloons_noisy.pgm
Image size is 640 x 480
Sequential calculation time : 3.074089 seconds
Parallel calculation time : 3.092431 seconds
Sequential time: 3.13
Parallel time: 3.14
Test PASSED

Input file is: data/bone_scint.pgm
Image size is 1129 x 1865
Sequential calculation time : 20.862417 seconds
Parallel calculation time : 22.377486 seconds
Sequential time: 21.20
Parallel time: 22.72
```

```
Test PASSED

Input file is: data/fuzzy.pgm
Image size is 564 x 770
Sequential calculation time : 4.763762 seconds
Parallel calculation time : 4.361592 seconds
Sequential time: 4.83
Parallel time: 4.43
Test PASSED

Input file is: data/lena512.pgm
Image size is 512 x 512
Sequential calculation time : 3.039198 seconds
Parallel calculation time : 2.603856 seconds
Sequential time: 3.08
Parallel time: 2.64
Test PASSED

Input file is: data/man.pgm
Image size is 1024 x 1024
Sequential calculation time : 10.353384 seconds
Parallel calculation time : 10.398058 seconds
Sequential time: 10.53
Parallel time: 10.57
Test PASSED

Input file is: data/Rainier_blur.pgm
Image size is 1920 x 1080
Sequential calculation time : 20.582173 seconds
Parallel calculation time : 21.640607 seconds
Sequential time: 20.98
Parallel time: 22.03
Test PASSED
```

Listing 13. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 1 niti sharpen

```
Input file is: data/balloons_noisy.pgm
Image size is 640 x 480
Sequential calculation time : 3.080412 seconds
Parallel calculation time : 1.526824 seconds
Sequential time: 3.18
Parallel time: 1.62
Test PASSED

Input file is: data/bone_scint.pgm
Image size is 1129 x 1865
Sequential calculation time : 20.853550 seconds
Parallel calculation time : 10.525402 seconds
Sequential time: 21.25
Parallel time: 10.95
Test PASSED

Input file is: data/fuzzy.pgm
Image size is 564 x 770
Sequential calculation time : 4.305402 seconds
Parallel calculation time : 2.163788 seconds
Sequential time: 4.41
Parallel time: 2.29
Test PASSED
```

```
Input file is: data/lena512.pgm
Image size is 512 x 512
Sequential calculation time : 2.645819 seconds
Parallel calculation time : 1.320546 seconds
Sequential time: 2.72
Parallel time: 1.42
Test PASSED

Input file is: data/man.pgm
Image size is 1024 x 1024
Sequential calculation time : 10.433358 seconds
Parallel calculation time : 5.239643 seconds
Sequential time: 10.67
Parallel time: 5.47
Test PASSED

Input file is: data/Rainier_blur.pgm
Image size is 1920 x 1080
Sequential calculation time : 20.507142 seconds
Parallel calculation time : 10.393989 seconds
Sequential time: 20.88
Parallel time: 11.10
Test PASSED
```

Listing 14. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 2 niti sharpen

```
Input file is: data/balloons_noisy.pgm
Image size is 640 x 480
Sequential calculation time : 3.045438 seconds
Parallel calculation time : 0.767659 seconds
Sequential time: 3.14
Parallel time: 0.88
Test PASSED

Input file is: data/bone_scint.pgm
Image size is 1129 x 1865
Sequential calculation time : 20.897869 seconds
Parallel calculation time : 6.678230 seconds
Sequential time: 21.23
Parallel time: 7.12
Test PASSED

Input file is: data/fuzzy.pgm
Image size is 564 x 770
Sequential calculation time : 4.958118 seconds
Parallel calculation time : 1.242398 seconds
Sequential time: 5.12
Parallel time: 1.31
Test PASSED

Input file is: data/lena512.pgm
Image size is 512 x 512
Sequential calculation time : 2.752969 seconds
Parallel calculation time : 0.657157 seconds
Sequential time: 2.91
Parallel time: 0.70
Test PASSED
```

```
Input file is: data/man.pgm
Image size is 1024 x 1024
Sequential calculation time : 11.347010 seconds
Parallel calculation time : 3.266164 seconds
Sequential time: 11.55
Parallel time: 3.49
Test PASSED

Input file is: data/Rainier_blur.pgm
Image size is 1920 x 1080
Sequential calculation time : 20.653484 seconds
Parallel calculation time : 5.179821 seconds
Sequential time: 21.03
Parallel time: 5.60
Test PASSED
```

Listing 15. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 4 niti sharpen

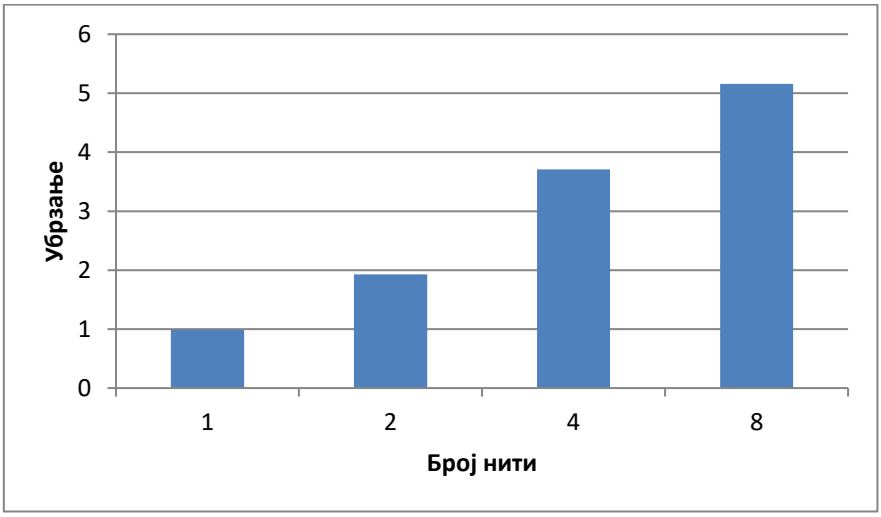
```
Input file is: data/balloons_noisy.pgm
Image size is 640 x 480
Sequential calculation time : 3.189433 seconds
Parallel calculation time : 0.607380 seconds
Sequential time: 3.28
Parallel time: 0.72
Test PASSED

Input file is: data/bone_scint.pgm
Image size is 1129 x 1865
Sequential calculation time : 21.232851 seconds
Parallel calculation time : 3.598527 seconds
Sequential time: 21.68
Parallel time: 4.02
Test PASSED

Input file is: data/fuzzy.pgm
Image size is 564 x 770
Sequential calculation time : 4.301108 seconds
Parallel calculation time : 0.751988 seconds
Sequential time: 4.42
Parallel time: 0.87
Test PASSED

Input file is: data/lena512.pgm
Image size is 512 x 512
Sequential calculation time : 2.593857 seconds
Parallel calculation time : 0.455599 seconds
Sequential time: 2.67
Parallel time: 0.54
Test PASSED

Input file is: data/man.pgm
Image size is 1024 x 1024
Sequential calculation time : 10.388198 seconds
Parallel calculation time : 1.818919 seconds
Sequential time: 10.61
Parallel time: 2.06
Test PASSED

Input file is: data/Rainier_blur.pgm
```

```
Image size is 1920 x 1080
Sequential calculation time : 20.635737 seconds
Parallel calculation time : 3.593528 seconds
Sequential time: 21.00
Parallel time: 3.98
Test PASSED
```

Listing 16. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 8 niti sharpen

4.3.2. Grafici ubrzanja

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



Slika 5. Grafik zavisnosti ubrzanja algoritma pomocu taskova od broja niti

4.3.3. Diskusija dobijenih rezultata

Paralelizacija je ubrzala izvršavanje za 2, 4 i 8 niti, a pri radu sa 1 niti nije usporila. Svaki piksel u suštini zavisi od susednih, ali na način na koji je implementiran zadatak, o tome ne moramo da brinemo, pošto se ne menja postojeća slika, već pravi nova matrica koja predstavlja izoštrenu sliku. Zbog toga ni u jednom trenutku nećemo upisati neodgovarajuću vrednost u podatak koji čitamo i nije potrebno međusobno isključenje niti. Najveće ubrzanje se dostiglo sa 8 niti - oko 5 puta, takođe primećujemo da ubrzanje nije puno zavisilo od dimenzije problema. U poglavlju 4.2.2. pomenute su dodatne for petlje koje je moguće paralelizovati – iako je paralelizacija u kodu koji je predat, nije uticala na rezultate. Paralelizacijom tih for petlji nije postignuto ni ubrzanje ni usporenje, međutim odlučili smo da ostane u kodu iz razloga što bi za neke značajno veće dimenzije problema moglo da bude od koristi.

5. PROBLEM 1 – MRI-GRIDDING

5.1. Tekst problema

Paralelizovati program koji vrši mapiranje neuniformnih podataka u 3D prostoru na regularnu mrežu u 3D prostoru. Svaka tačka iz neuniformnog 3D prostora doprinosi susednim tačkama u regularnoj mreži u skladu sa Kaiser-Bessel funkcijom za određivanje rastojanja. Program se nalazi u direktorijumu mri-gridding u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih su od interesa datoteke main.c i CPU_kernels.c. Analizirati dati kod i obratiti pažnju na način generisanja vrednosti tačaka u regularnoj mreži. Ukoliko je potrebno međusobno isključenje prilikom paralelizacije programa, koristiti dostupne OpenMP konstrukte. Obratiti pažnju na efikasnost međusobnog isključenja niti i po potrebi ga svesti na što je moguće manju meru uvođenjem pomoćnih struktura podataka. Ulazni test primeri se nalaze u direktorijumu data. Verifikaciju paralelizovanog rešenja vršiti nad nizovima gridData i sampleDensity iz glavnog programa. Način pokretanja programa se nalazi u datoteci run. [1, N]

5.2. Delovi koje treba paralelizovati

5.2.1. Diskusija

U funkciji gridding_Gold uočena je for petlja od 0 do n koju treba paralelizovati uzevši u obzir da je n veliki broj. Unutar ove for petlje, postoji još dosta forova koje je u principu moguće paralelizovati, ali se ta paralelizacija ne isplati. Prilikom rešavanja ovog zataka imali smo problem sa zadatim ACCURACY, međutim problem je bio do korišćenja float promenljivih i rešen je promenom sa float na double u strukturi cmplx.

5.2.2. Način paralelizacije

Paralelizacija je izvršena worksharing direktivom omp parallel for. Zanimljiva stvar je i to što je bilo potrebno napisati custom funkciju za redukciju nizova gridData i sampleDensity. Prvo smo pokušali da ti nizovi budu samo shared podatak i da upisivanje u njih bude atomic, ali nismo bili zadovoljni ubrzanjem pa smo napisali redukciju. Redukcija je popravila rezultat. Efikasnost redukcije najviše zavisi od parametra gridSize zato što za svaku nit treba da se inicijalizuje jedan veliki niz i posle da se sumiraju. Probali smo da smanjimo i povećamo veličinu 3D matrice u ulaznim podacima (inicijalno je bila 256x256x256) od koje taj parametar zavisi. Zaključili smo da se redukcija veoma isplati ukoliko je taj parametar manji (za 64x64x64 ubrzanje je bilo 4 puta sa redukcijom, a malo

manje od 2 puta bez), dok se veoma ne isplati ukoliko se poveća (za 512x512x512 rešenje redukcijom je sporije od sekvencijalnog, dok paralelno rešenje sa shared podacima je idalje oko 2 puta brže). Iz tog razloga u rešenju je implementirano da radi redukciju ukoliko je parametar gridSize dovoljno mali, a u suprotnom samo deli promenljive.

5.3. Rezultati

Vremena sekvencijalnog i paralelnog izvrsavanja u tabeli su srednje vrednosti 5 pokretanja programa sa zadatim parametrima i navedeni su u formatu (vreme sekvencijalnog izvršavanja) / (vreme paralelnog izvršavanja) = ubrzanje.

broj niti / ulazi	1	2	4	8
small.uks	1.956/2.112 = 0.9	1.978/1.312 = 1.5	1.964/0.978 = 2.0	1.962/0.984 = 1.9

5.3.1. Logovi izvršavanja

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

```
Reading parameters
  Number of samples = 2655910
  Grid Size = 256x256x256
  Input Matrix Size = 60x60x60
  Recon Matrix Size = 60x60x60
  Kernel Width = 5.000000
  KMax = 150.00 150.00 150.00
  Oversampling = 5.000000
  GPU Binsize = 32
  Use LUT = Yes
Reading input data from files
Generating Look-Up Table
N is 2655910
Sequential time: 1.93
Parallel time: 2.06
Test PASSED
```

Listing 17. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 1 niti mri-gridding

```
Reading parameters
  Number of samples = 2655910
  Grid Size = 256x256x256
  Input Matrix Size = 60x60x60
  Recon Matrix Size = 60x60x60
  Kernel Width = 5.000000
  KMax = 150.00 150.00 150.00
  Oversampling = 5.000000
  GPU Binsize = 32
```

```
  Use LUT = Yes
Reading input data from files
Generating Look-Up Table
N is 2655910
Sequential time: 1.94
Parallel time: 1.28
Test PASSED
```

Listing 18. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 2 niti mri-gridding

```
Reading parameters
  Number of samples = 2655910
  Grid Size = 256x256x256
  Input Matrix Size = 60x60x60
  Recon Matrix Size = 60x60x60
  Kernel Width = 5.000000
  KMax = 150.00 150.00 150.00
  Oversampling = 5.000000
  GPU Binsize = 32
  Use LUT = Yes
Reading input data from files
Generating Look-Up Table
N is 2655910
Sequential time: 1.94
Parallel time: 0.95
Test PASSED
```

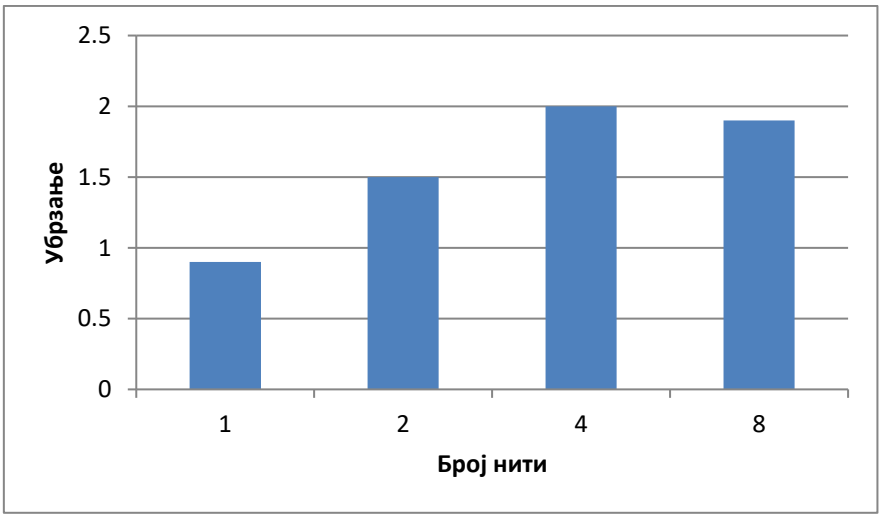
Listing 19. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 4 niti mri-gridding

```
Reading parameters
  Number of samples = 2655910
  Grid Size = 256x256x256
  Input Matrix Size = 60x60x60
  Recon Matrix Size = 60x60x60
  Kernel Width = 5.000000
  KMax = 150.00 150.00 150.00
  Oversampling = 5.000000
  GPU Binsize = 32
  Use LUT = Yes
Reading input data from files
Generating Look-Up Table
N is 2655910
Sequential time: 1.98
Parallel time: 0.97
Test PASSED
```

Listing 20. Poređenje sekvencijalnog izvršavanja i izvršavanja sa 8 niti mri-gridding

5.3.2. Grafici ubrzanja

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



Slika 6. Grafik zavisnosti ubrzanja naivnog algoritma od broja niti

5.3.3. Diskusija dobijenih rezultata

Paralelizacija je ubrzala izvršavanje za 2, 4 i 8 niti, a pri radu sa 1 niti došlo je do usporenja. Svaka tačka iz neuniformnog 3D prostora doprinosi susednim tačkama u regularnoj mreži u skladu sa Kaiser-Bessel funkcijom za određivanje rastojanja, pa je za korektno funkcionisanje potrebno međusobno isključenje niti. Dugo moguće rešenje je redukcija, tj kreiranje privatne kopije nizova koje računamo i posle ih spojiti. U sekciji 5.2.2. objašnjene su naše odluke oko korišćenja redukcije odnosno međusobnog isključenja u zavisnosti od veličine problema.