

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



DOMAĆI ZADATAK 1 – OPENMP

Izveštaj o urađenom domaćem zadatku

Predmetni asistent:

doc. dr Marko Mišić

Kandidat:

Aleksandar Ivanović 2013/0010

Beograd, april 2019.

SADRŽAJ

SADRŽAJ	2
1.PROBLEM 1 - SGEMM	4
1.1. TEKST PROBLEMA	4
1.2. DELOVI KOJE TREBA PARALELIZOVATI	4
1.2.1. <i>Diskusija</i>	4
1.2.2. <i>Način paralelizacije</i>	4
1.3. REZULTATI	5
1.3.1. <i>Logovi izvršavanja</i>	5
1.3.2. <i>Grafici ubrzanja</i>	11
1.3.3. <i>Diskusija dobijenih rezultata</i>	12
1.PROBLEM 2 - JACOBI	13
1.1. TEKST PROBLEMA	13
1.2. DELOVI KOJE TREBA PARALELIZOVATI	13
1.2.1. <i>Diskusija</i>	13
1.2.2. <i>Način paralelizacije</i>	13
1.3. REZULTATI	14
1.3.1. <i>Logovi izvršavanja</i>	14
1.3.2. <i>Grafici ubrzanja</i>	17
1.3.3. <i>Diskusija dobijenih rezultata</i>	17
1.PROBLEM 3 - KMEANS	18
1.1. TEKST PROBLEMA	18
1.2. DELOVI KOJE TREBA PARALELIZOVATI	18
1.2.1. <i>Diskusija</i>	18
1.2.2. <i>Način paralelizacije</i>	18
1.3. REZULTATI	19
1.3.1. <i>Logovi izvršavanja</i>	19

1.3.2.	<i>Grafici ubrzanja</i>	
	25	
1.3.3.	<i>Diskusija dobijenih rezultata.....</i>	
	26	

1.PROBLEM 1 - SGEMM

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

1.1. Tekst problema

Paralelizovati program koji vrši jednostavno generalizovano množenje matrica u jednostrukoj preciznosti *Single precision floating General Matrix Multiply* (SGEMM). SGEMM operacije je definisana sledećom formom:

$$C \leftarrow \alpha \cdot A \cdot B + \beta \cdot C$$

Program se nalazi u datoteci `sgemm.cc` 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 datoteci *run*. [1, N]

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

U okviru ove sekcije će biti opisani uočeni delovi koda koje je moguće paralelizovati. Biće diskutovati delovi nad kojima je izvršena paralelizacija. Ukoliko je neki deo moguće paralelizovati, ali to nije učinjeno, navesti razloge.

1.2.2. Način paralelizacije

Paralelizacija ovog problema je odrađena na tri načina:

1. Bez direktiva za podelu posla (`no_worksharing`)
2. Koristeći direktive za podelu posla (`worksharing`)
3. Koristeći OpenMP poslove (`tasking`)

Pri prvom pristupu (`no_worksharing`) kreiramo `NUMBER_OF_THREADS` niti i svakoj niti dodelimo $M \cdot N / \text{NUMBER_OF_THREADS}$ elemenata da ih obradi. Kako ukupan broj elemenata ne mora biti deljiv sa brojem niti prvoj (master) niti ćemo dodati ostatak elemenata. Nakon što

podelimo elemente nitima ostaje samo da se ručno preračunaju koordinate elemata matrica A i B koji će biti korišćeni u računanju odgovarajućeg elementa matrice C.

U drugom pristupu (worksharing) koristimo direktivu `#pragma omp parallel for` i optimizaciju `collapse(2)` kako bi smo zajedno paralelizovali petlje koje generišu elemente matrice C.

```
#pragma omp parallel for private(mm, nn, i) shared(A, B, C) collapse(2)

for (mm = 0; mm < m; ++mm) {

    for (nn = 0; nn < n; ++nn) {

        ...

    }

}
```

U trećem pristupu (tasking) koristimo algoritam iz prvog pristupa (no_worksharing) gde prvo podelimo posao nitima a onda kreiramo taskove unutar niti tako što je računanje svakog elementa koji pripada datoj niti zaseban task.

1.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 1.

1.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti. Obavezno uključiti u ispis i vremena izvršavanja. Logove pojedinačno uokviriti i obeležiti.

```
Opening file:data/small/input/matrix1.txt
Matrix dimension: 128x96
Opening file:data/small/input/matrix2t.txt
Matrix dimension: 160x96
matA: data/small/input/matrix1.txt, matB: data/small/input/matrix2t.txt, matC:
result_small.txt
```

```
NUMBER_OF_THREADS: 1

serial_implementation: time_elapsed: 0.005874 seconds
Opening file:result_small.txt for write.
Matrix dimension: 128x160

no_worksharing: time_elapsed: 0.005820 seconds
Opening file:result_small_no_worksharing.txt for write.
Matrix dimension: 128x160
Test PASSED

worksharing: time_elapsed: 0.005876 seconds
Opening file:result_small_worksharing.txt for write.
Matrix dimension: 128x160
Test PASSED

tasking: time_elapsed: 0.006124 seconds
Opening file:result_small_tasking.txt for write.
Matrix dimension: 128x160
Test PASSED

Opening file:data/medium/input/matrix1.txt
Matrix dimension: 1024x992
Opening file:data/medium/input/matrix2t.txt
Matrix dimension: 1056x992
matA: data/medium/input/matrix1.txt, matB: data/medium/input/matrix2t.txt, matC:
result_medium.txt

NUMBER_OF_THREADS: 1

serial_implementation: time_elapsed: 11.371906 seconds
Opening file:result_medium.txt for write.
Matrix dimension: 1024x1056

no_worksharing: time_elapsed: 10.911203 seconds
Opening file:result_medium_no_worksharing.txt for write.
Matrix dimension: 1024x1056
Test PASSED

worksharing: time_elapsed: 11.012757 seconds
```

```
Opening file:result_medium_worksharing.txt for write.  
Matrix dimension: 1024x1056  
Test PASSED
```

```
tasking: time_elapsed: 10.984793 seconds  
Opening file:result_medium_tasking.txt for write.  
Matrix dimension: 1024x1056  
Test PASSED
```

Listing 1. SGEMM izvršavanje sa jednom niti

```
Opening file:data/small/input/matrix1.txt  
Matrix dimension: 128x96  
Opening file:data/small/input/matrix2t.txt  
Matrix dimension: 160x96  
matA: data/small/input/matrix1.txt, matB: data/small/input/matrix2t.txt, matC:  
result_small.txt  
  
NUMBER_OF_THREADS: 2  
  
serial_implementation: time_elapsed: 0.007957 seconds  
Opening file:result_small.txt for write.  
Matrix dimension: 128x160  
  
no_worksharing: time_elapsed: 0.002962 seconds  
Opening file:result_small_no_worksharing.txt for write.  
Matrix dimension: 128x160  
Test PASSED  
  
worksharing: time_elapsed: 0.002964 seconds  
Opening file:result_small_worksharing.txt for write.  
Matrix dimension: 128x160  
Test PASSED  
  
tasking: time_elapsed: 0.003110 seconds  
Opening file:result_small_tasking.txt for write.  
Matrix dimension: 128x160  
Test PASSED  
  
Opening file:data/medium/input/matrix1.txt
```

```

Matrix dimension: 1024x992
Opening file:data/medium/input/matrix2t.txt
Matrix dimension: 1056x992
matA: data/medium/input/matrix1.txt, matB: data/medium/input/matrix2t.txt, matC:
result_medium.txt

NUMBER_OF_THREADS: 2

serial_implementation: time_elapsed: 10.956445 seconds
Opening file:result_medium.txt for write.
Matrix dimension: 1024x1056

no_worksharing: time_elapsed: 5.434174 seconds
Opening file:result_medium_no_worksharing.txt for write.
Matrix dimension: 1024x1056
Test PASSED

worksharing: time_elapsed: 5.487969 seconds
Opening file:result_medium_worksharing.txt for write.
Matrix dimension: 1024x1056
Test PASSED

tasking: time_elapsed: 5.467183 seconds
Opening file:result_medium_tasking.txt for write.
Matrix dimension: 1024x1056
Test PASSED

```

Listing 2. SGEMM izvršavanje sa dve niti

```

Opening file:data/small/input/matrix1.txt
Matrix dimension: 128x96
Opening file:data/small/input/matrix2t.txt
Matrix dimension: 160x96
matA: data/small/input/matrix1.txt, matB: data/small/input/matrix2t.txt, matC:
result_small.txt

NUMBER_OF_THREADS: 4

serial_implementation: time_elapsed: 0.006797 seconds
Opening file:result_small.txt for write.

```


Matrix dimension: 128x160

no_worksharing: time_elapsed: 0.001558 seconds

Opening file:result_small_no_worksharing.txt for write.

Matrix dimension: 128x160

Test PASSED

worksharing: time_elapsed: 0.001571 seconds

Opening file:result_small_worksharing.txt for write.

Matrix dimension: 128x160

Test PASSED

tasking: time_elapsed: 0.001760 seconds

Opening file:result_small_tasking.txt for write.

Matrix dimension: 128x160

Test PASSED

Opening file:data/medium/input/matrix1.txt

Matrix dimension: 1024x992

Opening file:data/medium/input/matrix2t.txt

Matrix dimension: 1056x992

matA: data/medium/input/matrix1.txt, matB: data/medium/input/matrix2t.txt, matC:
result_medium.txt

NUMBER_OF_THREADS: 4

serial_implementation: time_elapsed: 10.963845 seconds

Opening file:result_medium.txt for write.

Matrix dimension: 1024x1056

no_worksharing: time_elapsed: 2.734605 seconds

Opening file:result_medium_no_worksharing.txt for write.

Matrix dimension: 1024x1056

Test PASSED

worksharing: time_elapsed: 2.746316 seconds

Opening file:result_medium_worksharing.txt for write.

Matrix dimension: 1024x1056

Test PASSED

```
tasking: time_elapsed: 2.750091 seconds
Opening file:result_medium_tasking.txt for write.
Matrix dimension: 1024x1056
Test PASSED
```

Listing 3. SGEMM izvršavanje sa četiri niti

```
Opening file:data/small/input/matrix1.txt
Matrix dimension: 128x96
Opening file:data/small/input/matrix2t.txt
Matrix dimension: 160x96
matA: data/small/input/matrix1.txt, matB: data/small/input/matrix2t.txt, matC:
result_small.txt

NUMBER_OF_THREADS: 8

serial_implementation: time_elapsed: 0.006256 seconds
Opening file:result_small.txt for write.
Matrix dimension: 128x160

no_worksharing: time_elapsed: 0.002494 seconds
Opening file:result_small_no_worksharing.txt for write.
Matrix dimension: 128x160
Test PASSED

worksharing: time_elapsed: 0.001589 seconds
Opening file:result_small_worksharing.txt for write.
Matrix dimension: 128x160
Test PASSED

tasking: time_elapsed: 0.001984 seconds
Opening file:result_small_tasking.txt for write.
Matrix dimension: 128x160
Test PASSED

Opening file:data/medium/input/matrix1.txt
Matrix dimension: 1024x992
Opening file:data/medium/input/matrix2t.txt
Matrix dimension: 1056x992
```

```

matA: data/medium/input/matrix1.txt, matB: data/medium/input/matrix2t.txt, matC:
result_medium.txt

NUMBER_OF_THREADS: 8

serial_implementation: time_elapsed: 10.939976 seconds
Opening file:result_medium.txt for write.
Matrix dimension: 1024x1056

no_worksharing: time_elapsed: 2.719429 seconds
Opening file:result_medium_no_worksharing.txt for write.
Matrix dimension: 1024x1056
Test PASSED

worksharing: time_elapsed: 2.699889 seconds
Opening file:result_medium_worksharing.txt for write.
Matrix dimension: 1024x1056
Test PASSED

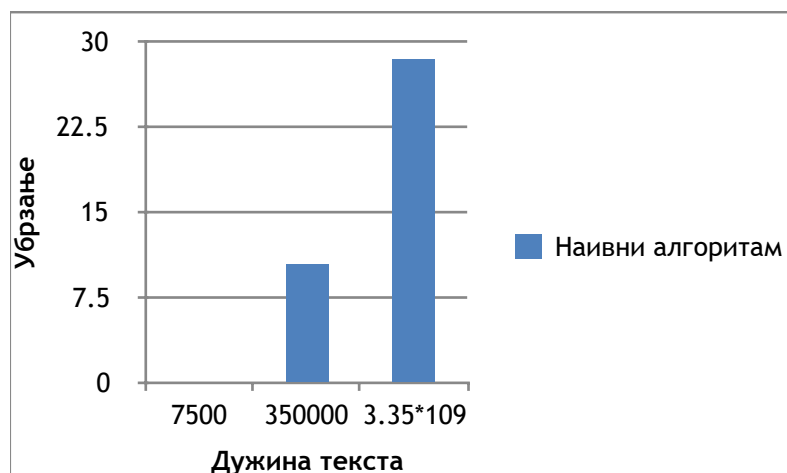
tasking: time_elapsed: 2.727060 seconds
Opening file:result_medium_tasking.txt for write.
Matrix dimension: 1024x1056
Test PASSED

```

Listing 4. SGEMM izvršavanje sa osam niti

1.3.2. Grafici ubrzanja

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



Slika 1. Grafik zavisnosti ubrzanja naivnog algoritma od dužine teksta za N = 4 niti (primer)

1.3.3. Diskusija dobijenih rezultata

Dobijene rezultate treba izdiskutovati i objasniti sa nekoliko rečenica u odnosu na dobijena ubrzanja (ili usporenja), način paralelizacije, karakteristike problema i sl.

1.PROBLEM 2 - JACOBI

U okviru ovog poglavlja je dat kratak izveštaj u vezi rešenja zadatog problema 2.

1.1. Tekst problema

Paralelizovati program koji rešava sistem linearnih jednačina $A * x = b$ Jakobijevim metodom. Kod koji treba paralelizovati se nalazi u datoteci jacobi.c u arhivi koja je priložena uz ovaj dokument. 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]

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

U okviru ove sekcije će biti opisani uočeni delovi koda koje je moguće paralelizovati. Biće diskutovati delovi nad kojima je izvršena paralelizacija. Ukoliko je neki deo moguće paralelizovati, ali to nije učinjeno, navesti razloge.

1.2.2. Način paralelizacije

U ovom zadatku su primenjene razne optimizacije paralelizacije. Prvo je urađena paralelizacija inicijalizacije niza:

```
#pragma omp parallel private(i)
{
    pragma omp for
    for (i = 0; i < n; i++) {
        b[i] = 0.0;
    }
    #pragma omp single
    b[n - 1] = (double) (n + 1);
```

```
#pragma omp for

for (i = 0; i < n; i++) {

    x[i] = 0.0;

}

}
```

1.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 1.

1.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti. Obavezno uključiti u ispis i vremena izvršavanja. Logove pojedinačno uokviriti i obeležiti.

```
NUMBER_OF_THREADS: 1
  Number of variables  N = 500
  Number of iterations M = 5000
serial_implementation: time_elapsed: 0.055901 seconds
parallel_implementation: time_elapsed: 0.054813 seconds
Test PASSED

NUMBER_OF_THREADS: 1
  Number of variables  N = 1000
  Number of iterations M = 10000
serial_implementation: time_elapsed: 0.200049 seconds
parallel_implementation: time_elapsed: 0.212172 seconds
Test PASSED

NUMBER_OF_THREADS: 1
  Number of variables  N = 5000
  Number of iterations M = 50000
serial_implementation: time_elapsed: 5.203035 seconds

parallel_implementation: time_elapsed: 5.365179 seconds
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 1
```

```
Number of variables N = 10000
```

```
Number of iterations M = 100000
```

```
serial_implementation: time_elapsed: 21.695650 seconds
```

```
parallel_implementation: time_elapsed: 22.029166 seconds
```

```
Test PASSED
```

Listing 1. JACOBI izvršavanje sa jednom niti

```
NUMBER_OF_THREADS: 2
```

```
Number of variables N = 500
```

```
Number of iterations M = 5000
```

```
serial_implementation: time_elapsed: 0.052274 seconds
```

```
parallel_implementation: time_elapsed: 0.032338 seconds
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
Number of variables N = 1000
```

```
Number of iterations M = 10000
```

```
serial_implementation: time_elapsed: 0.200767 seconds
```

```
parallel_implementation: time_elapsed: 0.115412 seconds
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
Number of variables N = 5000
```

```
Number of iterations M = 50000
```

```
serial_implementation: time_elapsed: 5.413749 seconds
```

```
parallel_implementation: time_elapsed: 2.860804 seconds
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
Number of variables N = 10000
```

```
Number of iterations M = 100000
```

```
serial_implementation: time_elapsed: 25.367626 seconds
```

```
parallel_implementation: time_elapsed: 15.560250 seconds
```

```
Test PASSED
```

Listing 2. JACOBI izvršavanje sa dve niti

```

NUMBER_OF_THREADS: 4
    Number of variables  N = 500
    Number of iterations M = 5000
serial_implementation: time_elapsed: 0.049346 seconds
parallel_implementation: time_elapsed: 0.020503 seconds
Test PASSED

NUMBER_OF_THREADS: 4
    Number of variables  N = 1000
    Number of iterations M = 10000
serial_implementation: time_elapsed: 0.197811 seconds
parallel_implementation: time_elapsed: 0.081883 seconds
Test PASSED

NUMBER_OF_THREADS: 4
    Number of variables  N = 5000
    Number of iterations M = 50000
serial_implementation: time_elapsed: 5.379560 seconds
parallel_implementation: time_elapsed: 1.609326 seconds
Test PASSED

NUMBER_OF_THREADS: 4
    Number of variables  N = 10000
    Number of iterations M = 100000
serial_implementation: time_elapsed: 21.550052 seconds
parallel_implementation: time_elapsed: 8.183444 seconds
Test PASSED

```

Listing 3. JACOBI izvršavanje sa četiri niti

```

NUMBER_OF_THREADS: 8
    Number of variables  N = 500
    Number of iterations M = 5000
serial_implementation: time_elapsed: 0.049212 seconds
parallel_implementation: time_elapsed: 0.024401 seconds
Test PASSED

NUMBER_OF_THREADS: 8

```



```

Number of variables N = 1000
Number of iterations M = 10000
serial_implementation: time_elapsed: 0.203817 seconds
parallel_implementation: time_elapsed: 0.068580 seconds
Test PASSED

NUMBER_OF_THREADS: 8
Number of variables N = 5000
Number of iterations M = 50000
serial_implementation: time_elapsed: 5.284178 seconds
parallel_implementation: time_elapsed: 1.466880 seconds
Test PASSED

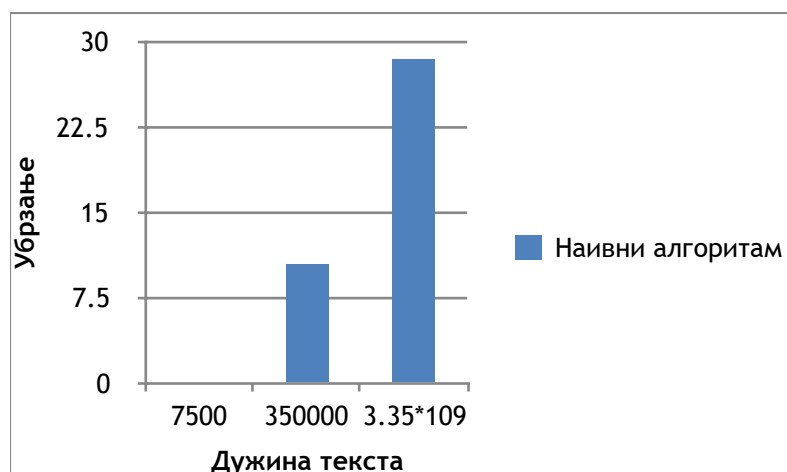
NUMBER_OF_THREADS: 8
Number of variables N = 10000
Number of iterations M = 100000
serial_implementation: time_elapsed: 21.635355 seconds
parallel_implementation: time_elapsed: 6.103708 seconds
Test PASSED

```

Listing 4. JACOBI izvršavanje sa osam niti

1.3.2. Grafici ubrzanja

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



Slika 1. Grafik zavisnosti ubrzanja naivnog algoritma od dužine teksta za N = 4 niti (primer)

1.3.3. Diskusija dobijenih rezultata

Dobijene rezultate treba izdiskutovati i objasniti sa nekoliko rečenica u odnosu na dobijena ubrzanja (ili usporenja), način paralelizacije, karakteristike problema i sl.

1.PROBLEM 3 - KMEANS

U okviru ovog poglavlja je dat kratak izveštaj u vezi rešenja zadatog problema 3.

1.1. Tekst problema

Paralelizovati program koji vrši k-means klasterizaciju podataka. Klasterizacija metodom k-srednjih vrednosti (eng. k-means clustering) je metod koji particioniše n objekata u k klastera u kojem svaki objekat pripada klasteru sa najbližom srednjom vrednošću. Objekat se sastoji od niza vrednosti - osobina (eng. features). Podelom objekata u potklaster, algoritam predstavlja sve objekte pomoću njihovi srednjih vrednosti (tzv. centroida potklastera). Inicijalni centroid za svaki potklaster se bira ili nasumično ili pomoću odgovarajuće heuristike. U svakoj iteraciji, algoritam pridružuje svaki objekat najbližem centroidu na osnovu definisane metrike. Novi centroidi za sledeću iteraciju se izračunavaju usrednjavanjem svih objekata unutar potklastera. Algoritam se izvršava sve dok se makar jedan objekat pomera iz jednog u drugi potklaster. Program se nalazi u direktorijumu kmeans u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih su od interesa datoteke kmeans.c, cluster.c i kmeans_clustering.c. Analizirati dati kod i obratiti pažnju na generisanje novih centroida u svakoj iteraciji unutar datoteke kmeans_clustering.c. Ukoliko je potrebno međusobno isključenje prilikom paralelizacije programa, koristiti dostupne OpenMP konstrukte. Obratiti pažnju na efikasnost međusobnog isključenja niti i svesti ga na što je moguće manju meru uvođenjem pomoćnih struktura podataka. Ulazni test primeri se nalaze u direktorijumu data, a način pokretanja programa u datoteci run. [1, N]

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

U okviru ove sekcije će biti opisani uočeni delovi koda koje je moguće paralelizovati. Biće diskutovati delovi nad kojima je izvršena paralelizacija. Ukoliko je neki deo moguće paralelizovati, ali to nije učinjeno, navesti razloge.

1.2.2. Način paralelizacije

Paralelizacija ovog problema je odrađena na tri načina:

1. Bez direktiva za podelu posla (no_worksharing)

2. Koristeći direktive za podelu posla (worksharing)
3. Koristeći OpenMP poslove (tasking)

Pri prvom pristupu (no_worksharing) kreiramo NUMBER_OF_THREADS niti i svakoj niti dodelimo $M \cdot N / \text{NUMBER_OF_THREADS}$ elemenata da ih obradi. Kako ukupan broj elemenata ne mora biti deljiv sa brojem niti prvoj (master) niti ćemo dodati ostatak elemenata. Nakon što podelimo elemente nitima ostaje samo da se ručno preračunaju koordinate elemata matrica A i B koji će biti korišćeni u računanju odgovarajućeg elementa matrice C.

U drugom pristupu (worksharing) koristimo direktivu #pragma omp parallel for i optimizaciju collapse(2) kako bi smo zajedno paralelizovali petlje koje generišu elemente matrice C.

```
#pragma omp parallel for private(mm, nn, i) shared(A, B, C) collapse(2)

for (mm = 0; mm < m; ++mm) {
    for (nn = 0; nn < n; ++nn) {
        ...
    }
}
```

U trećem pristupu (tasking) koristimo algoritam iz prvog pristupa (no_worksharing) gde prvo podelimo posao nitima a onda kreiramo taskove unutar niti tako što je računanje svakog elementa koji pripada datoj niti zaseban task.

1.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 1.

1.3.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja za definisane test primere i različit broj niti. Obavezno uključiti u ispis i vremena izvršavanja. Logove pojedinačno uokviriti i obeležiti.

```
NUMBER_OF_THREADS: 1
I/O completed
number of Clusters 5
number of Attributes 34

Time for serial processing: 0.000975
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 0.001012

Test PASSED

NUMBER_OF_THREADS: 1
I/O completed

number of Clusters 5
number of Attributes 34

Time for serial processing: 10.313196
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 10.528897

Test PASSED

NUMBER_OF_THREADS: 1
I/O completed
number of Clusters 5
number of Attributes 34

Time for serial processing: 2.236829
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 2.268729
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 1
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 9.428226
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for OpenMP processing: 9.698274
```

```
Test PASSED
```

Listing 1. KMEANS izvršavanje sa jednom niti

```
NUMBER_OF_THREADS: 2
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 0.000479
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for OpenMP processing: 0.000327
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 10.316224
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for OpenMP processing: 5.963640
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 2.238284
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for OpenMP processing: 1.266606
```

```
Test PASSED
```

```
NUMBER_OF_THREADS: 2
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 9.445290
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for OpenMP processing: 5.395080
```

```
Test PASSED
```

Listing 2. KMEANS izvršavanje sa dve niti

```
NUMBER_OF_THREADS: 4
```

```
I/O completed
```

```
number of Clusters 5
```

```
number of Attributes 34
```

```
Time for serial processing: 0.000465
```

```
number of Clusters 5
```

number of Attributes 34

Time for OpenMP processing: 0.000261

Test PASSED

NUMBER_OF_THREADS: 4

I/O completed

number of Clusters 5

number of Attributes 34

Time for serial processing: 10.322841

number of Clusters 5

number of Attributes 34

Time for OpenMP processing: 3.845401

Test PASSED

NUMBER_OF_THREADS: 4

I/O completed

number of Clusters 5

number of Attributes 34

Time for serial processing: 2.234710

number of Clusters 5

number of Attributes 34

Time for OpenMP processing: 0.782745

Test PASSED

NUMBER_OF_THREADS: 4

I/O completed

number of Clusters 5

number of Attributes 34

Time for serial processing: 9.487083

number of Clusters 5

```
number of Attributes 34

Time for OpenMP processing: 4.872271

Test PASSED: 1024x1056
Test PASSED
```

Listing 3. KMEANS izvršavanje sa četiri niti

```
NUMBER_OF_THREADS: 8
I/O completed
number of Clusters 5
number of Attributes 34

Time for serial processing: 0.000473
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 0.017184

Test PASSED

NUMBER_OF_THREADS: 8
I/O completed
number of Clusters 5
number of Attributes 34

Time for serial processing: 10.306361
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 5.228033

Test PASSED

NUMBER_OF_THREADS: 8
I/O completed
number of Clusters 5
number of Attributes 34
```



```

Time for serial processing: 2.242113
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 0.965846

Test PASSED

NUMBER_OF_THREADS: 8
I/O completed
number of Clusters 5
number of Attributes 34

Time for serial processing: 9.468104
number of Clusters 5
number of Attributes 34

Time for OpenMP processing: 4.842897

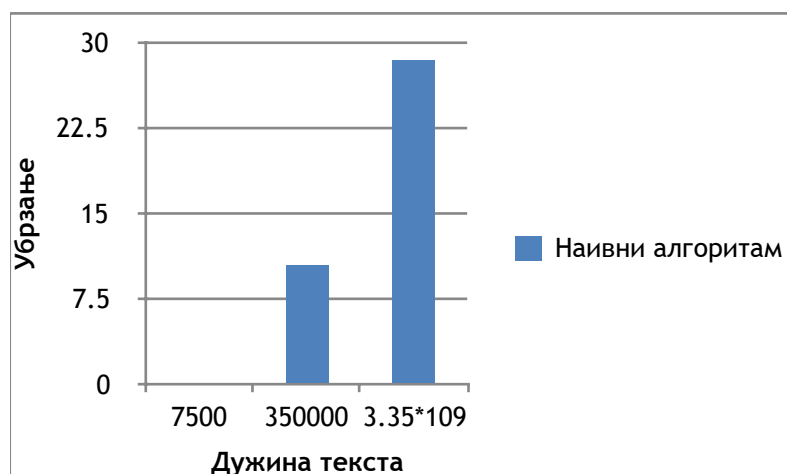
Test PASSED

```

Listing 4. KMEANS izvršavanje sa osam niti

1.3.2. Grafici ubrzanja

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



Slika 1. Grafik zavisnosti ubrzanja naivnog algoritma od dužine teksta za N = 4 niti (primer)

1.3.3. Diskusija dobijenih rezultata

Dobijene rezultate treba izdiskutovati i objasniti sa nekoliko rečenica u odnosu na dobijena ubrzanja (ili usporenja), način paralelizacije, karakteristike problema i sl.