

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



DOMAĆI ZADATAK 4 – CUDA

Izveštaj o urađenom domaćem zadatku

Predmetni saradnici:

doc. dr Marko Mišić

dipl. ing. Matija Dodović

Studenti:

Nemanja Mehović 2022/3088

Beograd, jun 2023.

SADRŽAJ

SADRŽAJ.....	2
1. PROBLEM 1 - PRIME.....	3
1.1. TEKST PROBLEMA.....	3
1.2. DELOVI KOJE TREBA PARALELIZOVATI	3
1.2.1. <i>Diskusija</i>	3
1.2.2. <i>Način paralelizacije</i>	3
1.3. REZULTATI	3
1.3.1. <i>Logovi izvršavanja</i>	3
1.3.2. <i>Grafici ubrzanja</i>	8
1.3.3. <i>Diskusija dobijenih rezultata</i>	8
2. PROBLEM 2 - FEYMAN.....	9
2.1. TEKST PROBLEMA.....	9
2.2. DELOVI KOJE TREBA PARALELIZOVATI	9
2.2.1. <i>Diskusija</i>	9
2.2.2. <i>Način paralelizacije</i>	9
2.3. REZULTATI	9
2.3.1. <i>Logovi izvršavanja</i>	9
2.3.2. <i>Grafici ubrzanja</i>	12
2.3.3. <i>Diskusija dobijenih rezultata</i>	13
3. PROBLEM 3 - MOLDYN	14
3.1. TEKST PROBLEMA.....	14
3.2. DELOVI KOJE TREBA PARALELIZOVATI	14
3.2.1. <i>Diskusija</i>	14
3.2.2. <i>Način paralelizacije</i>	15
3.3. REZULTATI	15
3.3.1. <i>Logovi izvršavanja</i>	15
3.3.2. <i>Grafici ubrzanja</i>	16
3.3.3. <i>Diskusija dobijenih rezultata</i>	16

1.PROBLEM 1 - PRIME

1.1. Tekst problema

Paralelizovati program koji vrši određivanje ukupnog broja prostih brojeva u zadatom opsegu. Prilikom zadavanja izvršne konfiguracije jezgra, koristiti 1D rešetku (grid). Obratiti pažnju na efikasnost paralelizacije i potrebu za redukcijom. Program se nalazi u datoteci prime.c u arhivi koja je priložena uz ovaj dokument. Program testirati sa parametrima koji su dati u datoteci run.

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

Postoji jedno mesto u programu koje ima smisla paralelizovati, a to je glavna for petlja u funkciji *prime_number*.

1.2.2. Način paralelizacije

Paralelizacija for petlje je urađena tako da svaka nit radi nad samo jednim brojom. Proveće koliko prostih brojeva postoji u tom opsegu i sačuvati u shared memoriju bloka. Nakon toga izvršiće se redukcija nad shared memorijom i rezultat će biti sačuvan u niz globalne memorije koji se prosleđuje procesoru za poslednju redukciju.

1.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 1.

1.3.1. Logovi izvršavanja

SEQUENTIAL EXECUTION		
17 June 2023 12:18:19 PM		
PRIME TEST		
Call PRIME_NUMBER to count the primes from 1 to N.		
N	Pi	Time
1	0	0.000002
2	1	0.000001
4	2	0.000001

8	4	0.000002
16	6	0.000001
32	11	0.000003
64	18	0.000007
128	31	0.000020
256	54	0.000057
512	97	0.000193
1024	172	0.000654
2048	309	0.002278
4096	564	0.008315
8192	1028	0.013283
16384	1900	0.018706
32768	3512	0.068290
65536	6542	0.254633
131072	12251	0.957078

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:21 PM

Execution time for sequential 1.323832 s

PARALLEL EXECUTION

17 June 2023 12:18:21 PM

PRIME TEST

Call PRIME_NUMBER to count the primes from 1 to N.

N	Pi	Time
1	0	0.074249
2	1	0.000087
4	2	0.000092
8	4	0.000091
16	6	0.000100
32	11	0.000098
64	18	0.000098
128	31	0.000109

256	54	0.000122
512	97	0.000147
1024	172	0.000220
2048	309	0.000405
4096	564	0.000741
8192	1028	0.001412
16384	1900	0.002799
32768	3512	0.006249
65536	6542	0.018138
131072	12251	0.060547

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:21 PM

Execution time for parallel 0.186596 s

Test PASSED

SEQUENTIAL EXECUTION

17 June 2023 12:18:21 PM

PRIME TEST

Call PRIME_NUMBER to count the primes from 1 to N.

N	Pi	Time
5	3	0.000001
50	15	0.000001
500	95	0.000031
5000	669	0.002041
50000	5133	0.153022
500000	41538	12.444000

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:34 PM

Execution time for sequential 12.599400 s

PARALLEL EXECUTION

17 June 2023 12:18:34 PM

PRIME TEST

Call PRIME_NUMBER to count the primes from 1 to N.

N	Pi	Time
5	3	0.049617
50	15	0.000098
500	95	0.000149
5000	669	0.000888
50000	5133	0.011792
500000	41538	0.720484

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:34 PM

Execution time for parallel 0.804510 s

Test PASSED

SEQUENTIAL EXECUTION

17 June 2023 12:18:34 PM

PRIME TEST

Call PRIME_NUMBER to count the primes from 1 to N.

N	Pi	Time
1	0	0.000000
4	2	0.000000
16	6	0.000001
64	18	0.000002

256	54	0.000010
1024	172	0.000113
4096	564	0.001452
16384	1900	0.018556
65536	6542	0.250147

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:35 PM

Execution time for sequential 0.271012 s

PARALLEL EXECUTION

17 June 2023 12:18:35 PM

PRIME TEST

Call PRIME_NUMBER to count the primes from 1 to N.

N	Pi	Time
1	0	0.042492
4	2	0.000093
16	6	0.000091
64	18	0.000094
256	54	0.000107
1024	172	0.000204
4096	564	0.000656
16384	1900	0.002494
65536	6542	0.016628

PRIME_TEST

Normal end of execution.

17 June 2023 12:18:35 PM

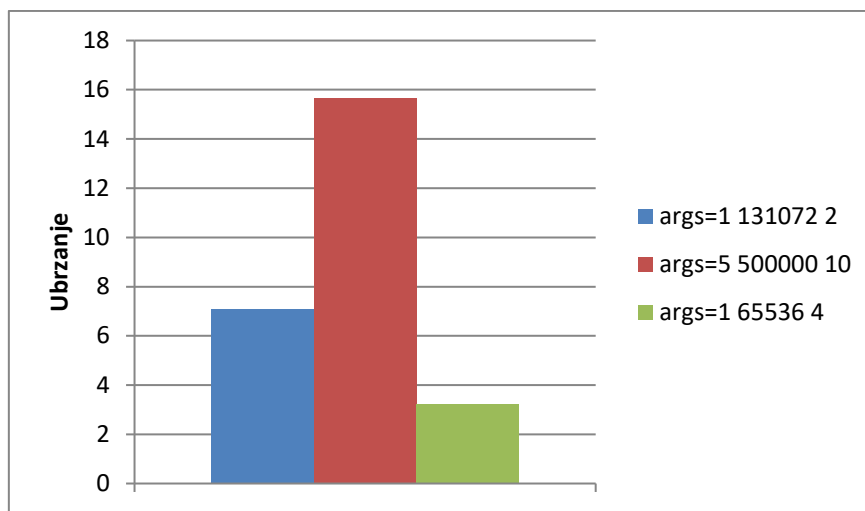
Execution time for parallel 0.083913 s

Test PASSED

Listing 1. Log izvršavanja svih test primera za problem 1

1.3.2. Grafici ubrzanja

U okviru ove sekcije je dat grafik ubrzanja u odnosu na sekvencijalnu implementaciju.



Slika 1. Grafik ubrzanja

1.3.3. Diskusija dobijenih rezultata

Paralelizacijom programa sa GPU nitima je dovelo do značajnog ubrzanja. Ovo se naročito viti kada je posao dovoljno veliki, gde će overhead koji se unosi za prenošenje podatka sa CPU-a na GPU biti zanemarljiv zbog količine podataka koji se obrađuju, što možemo i videti na grafu.

2.PROBLEM 2 - FEYMAN

2.1. Tekst problema

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

2.2. Delovi koje treba paralelizovati

2.2.1. Diskusija

Za razliku od prethodnog problema gde je mogućnost paralelizacije bila samo na jednom mestu kod ovog problema imamo četiri mesta u kodu koja je moguće paralelizovati. Te lokacije su ugnježdene for i while petlje u main funkciji. Za paralelizaciju izabrao sam unutrašnju for petlju zato što obrađuje najveći deo posla i jedina je sa nepoznatim brojem iteracija.

2.2.2. Način paralelizacije

Paralelizacija je implementirana na sličan način kao i u prethodnom problemu. Prilikom paralelizacije je vođeno posbno računa da se nasumičan deo algoritma idalje ostvari. Ovo je postignuto tako što je seed za svaki GPU thread bio promenjen, korišćenjem njegovog id-a .

2.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 2.

2.3.1. Logovi izvršavanja

```
SEQUENTIAL EXECUTION
17 June 2023 12:21:57 PM
A = 3.000000
B = 2.000000
C = 1.000000
N = 1000
H = 0.0010
```

RMS absolute error in solution = 2.171700e-02

17 June 2023 12:22:20 PM

Execution time for sequential 23.128995 s

PARALLEL EXECUTION

17 June 2023 12:22:20 PM

A = 3.000000

B = 2.000000

C = 1.000000

N = 1000

H = 0.0010

RMS absolute error in solution = 2.138846e-02

17 June 2023 12:24:50 PM

Execution time for parallel 150.444095 s

Test PASSED

SEQUENTIAL EXECUTION

17 June 2023 12:24:50 PM

A = 3.000000

B = 2.000000

C = 1.000000

N = 5000

H = 0.0010

RMS absolute error in solution = 2.127277e-02

17 June 2023 12:26:47 PM

Execution time for sequential 116.750988 s

PARALLEL EXECUTION

17 June 2023 12:26:47 PM

A = 3.000000

B = 2.000000

C = 1.000000

N = 5000

H = 0.0010

RMS absolute error in solution = 2.113288e-02

17 June 2023 12:29:31 PM

Execution time for parallel 163.796232 s

Test PASSED

SEQUENTIAL EXECUTION

17 June 2023 12:29:31 PM

A = 3.000000

B = 2.000000

C = 1.000000

N = 10000

H = 0.0010

RMS absolute error in solution = 2.109998e-02

17 June 2023 12:33:26 PM

Execution time for sequential 235.007716 s

PARALLEL EXECUTION

17 June 2023 12:33:26 PM

A = 3.000000

B = 2.000000

C = 1.000000

N = 10000

H = 0.0010

RMS absolute error in solution = 2.120956e-02

17 June 2023 12:36:15 PM

Execution time for parallel 169.160446 s

Test PASSED

SEQUENTIAL EXECUTION

17 June 2023 12:36:15 PM

```
A = 3.000000
B = 2.000000
C = 1.000000
N = 20000
H = 0.0010

RMS absolute error in solution = 2.102653e-02
17 June 2023 12:44:08 PM
Execution time for sequential 472.874443 s

PARALLEL EXECUTION
17 June 2023 12:44:08 PM
A = 3.000000
B = 2.000000
C = 1.000000
N = 20000
H = 0.0010

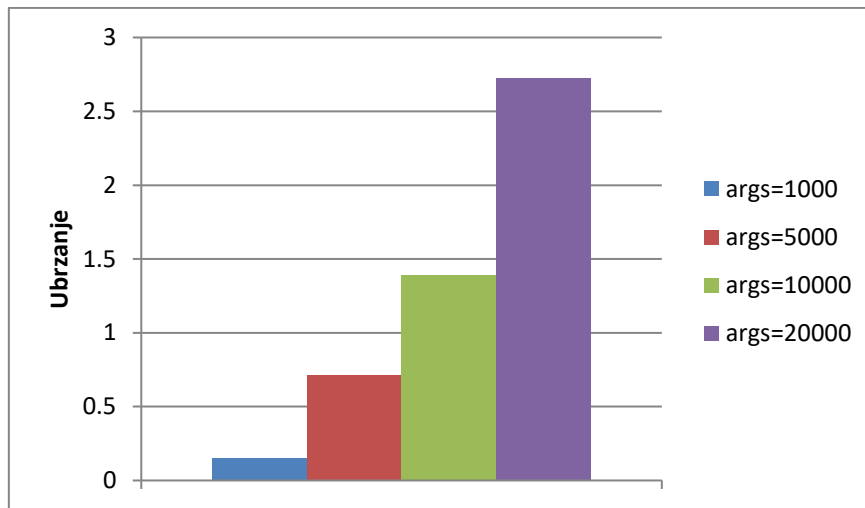
RMS absolute error in solution = 2.098655e-02
17 June 2023 12:47:01 PM
Execution time for parallel 173.517990 s

Test PASSED
```

Listing 2. Log izvršavanja svih test primera za problem 2

2.3.2. Grafici ubrzanja

U okviru ove sekcije je dat grafik ubrzanja u odnosu na sekvencijalnu implementaciju.



Slika 2. Grafik ubrzanja

2.3.3. Diskusija dobijenih rezultata

Za razliku od prethodnog zadatka paralelizacijom ovog problema korišćenjem GPU-a nije dovelo do prevelikog ubrzanja. Idalje je postojalo ubrzanje kada je bila dovoljno velika količina podataka da se obradi, ali čak i u tim situacijama ubrzanje nije bilo preveliko. Razlog za ovo jeste priroda samog problema koja sadrži ogromnu količinu grananja koje je loše za izvršavanje na GPU. Ovaj faktor plus dodatan overhead koji se dobije za prenošenje podataka sa CPU-a na GPU je čak i dovelo do nezanemarljivog usporenja za obradu male količine podataka.

3. PROBLEM 3 - MOLDYN

3.1. Tekst problema

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

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

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

3.2. Delovi koje treba paralelizovati

3.2.1. Diskusija

Problem sadrži prilično veliku količinu lokacija koje je moguće paralelizovati nezavisno jedan od drugog, ali glavni deo koji trebamo paralelizovati nalazi se u funkciji `forces` koja nas košta najveću količinu vremena prilikom izvršavanja.

3.2.2. Način paralelizacije

Paralelizacija je urađena na sličan način kao i u prethodna dva problema gde se radila redukcija za shared promenljive epot i vir, ali pored redukcije sada su uvedene i atomične operacije nad glavnim nizom koji obrađujemo zato što se nalazi u globalnoj memoriji i ne postoji jednostavan način da se omogućiti rađanje sa ne globalnim podacima u njegovom slučaju.

3.3. Rezultati

U okviru ove sekcije su izloženi rezultati paralelizacije problema 3.

3.3.1. Logovi izvršavanja

```
SEQUENTIAL EXECUTION
Molecular Dynamics Simulation example program
-----
number of particles is ..... 13500
side length of the box is ..... 25.323179
cut off is ..... 3.750000
reduced temperature is ..... 0.722000
basic timestep is ..... 0.064000
temperature scale interval ..... 10
stop scaling at move ..... 20
print interval ..... 5
total no. of steps ..... 20

  i      ke      pe      e      temp      pres      vel      rp
  ----  -
    5 12619.1758 -91985.3542 -79366.1784  0.6232  -5.2880  0.1821  39.7
   10 14619.4170 -86181.5919 -71562.1749  0.7220  -2.8265  0.1336  14.1
   15 11405.1707 -82966.3254 -71561.1547  0.5633  -1.5094  0.1714  33.6
   20 10825.0423 -82385.8646 -71560.8222  0.5346  -1.2219  0.1679  32.2
Time = 10.721217

PARALLEL EXECUTION
Molecular Dynamics Simulation example program
-----
number of particles is ..... 13500
side length of the box is ..... 25.323179
cut off is ..... 3.750000
reduced temperature is ..... 0.722000
```

basic timestep is	0.064000
temperature scale interval	10
stop scaling at move	20
print interval	5
total no. of steps	20

i	ke	pe	e	temp	pres	vel	rp
-----	-----	-----	-----	-----	-----	-----	-----
5	12619.1758	-91985.3542	-79366.1784	0.6232	-5.2880	0.1821	39.7
10	14619.4170	-86181.5919	-71562.1749	0.7220	-2.8265	0.1336	14.1
15	11405.1707	-82966.3254	-71561.1547	0.5633	-1.5094	0.1714	33.6
20	10825.0423	-82385.8646	-71560.8222	0.5346	-1.2219	0.1679	32.2

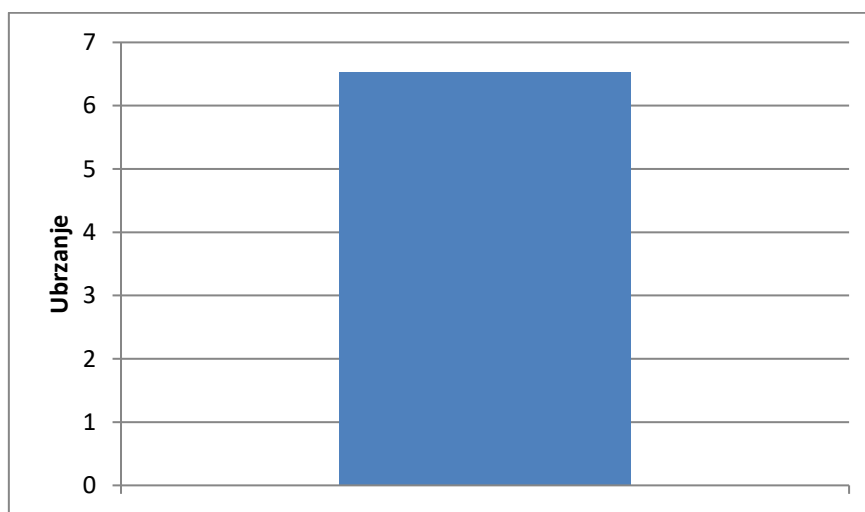
Time = 1.643232

Test PASSED

Listing 3. Log izvršavanja problema 3

3.3.2. Grafici ubrzanja

U okviru ove sekcije je dat grafik ubrzanja u odnosu na sekvencijalnu implementaciju.



Slika 3. Grafik ubrzanja

3.3.3. Diskusija dobijenih rezultata

Možemo videti da je došlo do ubrzanja izvršavanja programa koristeći GPU za izračunavanje funkcije forces, ali ovo ubrzanje nije preveliko. Korišćenjem atomičnih operacija je značajno usporilo moguće ubrzanje samog programa.