

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



DOMAĆI ZADATAK 3 – CUDA

Izveštaj o urađenom domaćem zadatku

Predmetni saradnici:

doc. dr Marko Mišić

dipl. ing. Pavle Divović

Studenti:

Andrea Popović 2018/0316

Vukašin Nedeljković 2018/0217

Beograd, jun 2022.

SADRŽAJ

SADRŽAJ	2
1. PROBLEM 1 - SIMPLEX	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	4
1.3.1. <i>Logovi izvršavanja</i>	4
1.3.2. <i>Grafici ubrzanja</i>	5
1.3.3. <i>Diskusija dobijenih rezultata</i>	5
2. PROBLEM 2 – GAME OF LIFE	6
2.1. TEKST PROBLEMA.....	6
2.2. DELOVI KOJE TREBA PARALELIZOVATI	6
2.2.1. <i>Diskusija</i>	6
2.2.2. <i>Način paralelizacije</i>	7
2.3. REZULTATI	7
2.3.1. <i>Logovi izvršavanja</i>	7
2.3.2. <i>Grafici ubrzanja</i>	8
2.3.3. <i>Diskusija dobijenih rezultata</i>	8
3. PROBLEM 3 – HOTSPOT	9
3.1. TEKST PROBLEMA.....	9
3.2. DELOVI KOJE TREBA PARALELIZOVATI	9
3.2.1. <i>Diskusija</i>	9
3.2.2. <i>Način paralelizacije</i>	9
3.3. REZULTATI	10
3.3.1. <i>Logovi izvršavanja</i>	11
3.3.2. <i>Grafici ubrzanja</i>	12
3.3.3. <i>Diskusija dobijenih rezultata</i>	12

1. PROBLEM 1 - SIMPLEX

1.1. Tekst problema

Paralelizovati program koji računa integral funkcije F na osnovu unutrašnjosti simplex-a (<https://en.wikipedia.org/wiki/Simplex>) u 20 dimenzija korišćenjem Monte Carlo metode. Program se nalazi u datoteci `simplex.c`. U izvornom kodu data je matrica eksponenata jednačine i ivica simplexa. Ulazni parametar programa je broj iteracija aproksimacije. Program testirati sa parametrima koji su dati u datoteci `run`.

1.2. Delovi koje treba paralelizovati

1.2.1. Diskusija

Uočili smo par potencijalnih funkcija za paralelizaciju. Nakon njihove paralelizacije smo zaključili da nam najbolje performanse daje paralelizacija `simplex_unit_to_general` funkcije.

1.2.2. Način paralelizacije

Prvo je alocirana memorija na uređaju za potrebne nizove. Nakon toga smo pozivali kernel sa dimenzijama `numBlocks` (predstavlja broj n), $m*m$ (nismo koristili `NUM_OF_GPU_THREADS` zato što je $m*m$ dosta manje od `NUM_OF_GPU_THREADS` a nije nam potrebno toliko niti). U krenelu svaka nit računa novu vrednost i na kraju vrsimo redukciju i sabiranje svih elemenata.

1.3. Rezultati

ulaz	Ubrzanje
500000	$\frac{1130.017700}{545.342896}$ $= 2.07$
100000	$\frac{2367.957031}{1249.830933}$ $= 1.89$
1000000	$\frac{20738.814453}{10370.778320}$ $= 2.00$
Srednje ubrzanje	1.99

1.3.1. Logovi izvršavanja

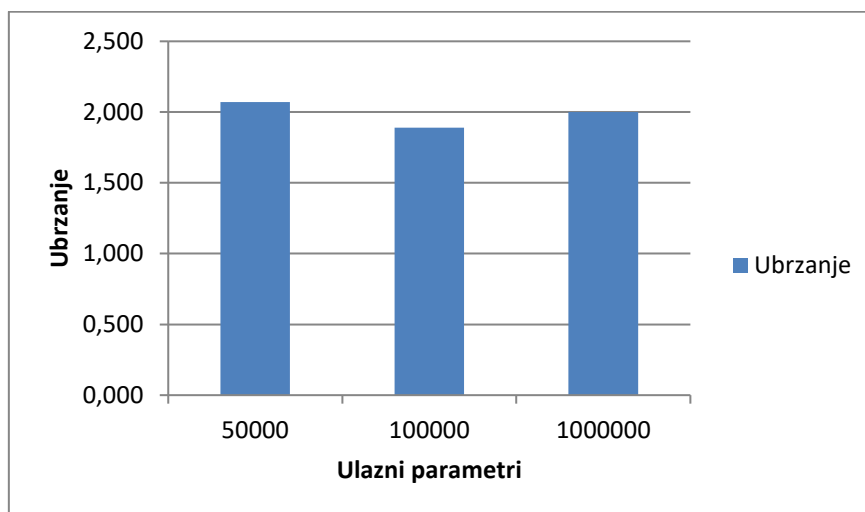
```
Simplex 50000
Time elapsed, sequential in ms: 1130.017700
Time elapsed, parallel in ms: 545.342896
Test PASSED

Simplex 100000
Time elapsed, sequential in ms: 2367.957031
Time elapsed, parallel in ms: 1249.830933
Test PASSED

Simplex 1000000
Time elapsed, sequential in ms: 20738.814453
Time elapsed, parallel in ms: 10370.778320
Test PASSED
```

Listing 1. Sekvencijalna i paralelna izvršavanja

1.3.2. Grafici ubrzanja



Slika 1. Grafik ubrzanja za različite ulazne parametre

1.3.3. Diskusija dobijenih rezultata

Primećujemo ubrzanja u odnosu na sekvencijalni kod. Ubrzanja nisu velika zato što su u problemu dimenzije matrica dosta male i nije moguće u potpunosti iskoristiti potencijal GPU. Usled numeričke nestabilnosti u radu sa realnim podacima imamo mala odstupanja koja su nešto veća od ACCURACY zbog neslaganja na centralnom i grafičkom procesoru.

2. PROBLEM 2 – GAME OF LIFE

2.1. Tekst problema

Paralelizovati program koji implementira simulaciju ćelijskog automata Game of Life. Simulacija je predstavljena dvodimenzionalnom matricom dimenzija $w \times h$, a svaka ćelija c može uzeti vrednost 1 ukoliko predstavlja živu ćeliju, a 0 ukoliko je mrtva. Za svaku ćeliju se vrši izračunavanje vrednosti n koja predstavlja zbir živih ćelija u susedstvu posmatrane ćelije. Posmatra se osam suseda. Ćelije se rađaju i umiru prema pravilima iz sledeće tabele.

Vrednost C	Vrednost N	Nova vrednost C	Komentar
1	0, 1	0	Usamljena ćelija umire
1	4, 5, 6, 7, 8	0	Ćelija umire usled prenaseljenosti
1	2,3	1	Ćelija živi
0	3	1	Rađa se nova ćelija
0	0, 1, 2, 4, 6, 7, 8	0	Nema promene stanja

Može se smatrati da su ćelije van opsega posmatrane matrice mrtve. Kod koji treba paralelizovati se nalazi u datoteci `gameoflife.c` u arhivi koja je priložena uz ovaj dokument. Program se može prevesti u dve konfiguracije: sa vizuelnim prikazom i bez vizuelnog prikaza, u zavisnosti da li je definisan makro `LIFE_VISUAL`. Prevođenje sa vizuelnim prikazom se može izvršiti naredbom `make visual`. Paralelizovati konfiguraciju bez vizuelnog prikaza, a vreme meriti na nivou cele simulacije i na nivou jednog izvršavanja funkcije `evolve`. Koristiti 2D organizaciju jezgra, ukoliko je moguće. Program testirati sa parametrima koji su dati u datoteci `run`

2.2. Delovi koje treba paralelizovati

2.2.1. Diskusija

Paralelizaciju je jedino moguće izvršiti u funkciji `evolve`. U funkciji postoje po 3 ugneždene `for` petlje. Poslednju ugneždenu petlju se ne isplati paralelizovati zato što se u njoj jedino izvršava dodela vrednosti, dok druga ugneždjena petlja ima mali broj iteracija i nju smo optimizovali korišćenjem tehnike `loop unrolling`. Zbog toga kompletnu paralelizaciju vršimo nad prvom ugneždenom petljom.

2.2.2. Način paralelizacije

Prvo je alocirana memorija na uređaju za potrebne matrice. Nakon toga smo pozivali kernel sa dimenzijama numBlocks (predstavlja gornji ceo deo od dimenzije matrice podeljene sa ukupnim brojem niti po bloku), NUM_OF_GPU_THREADS. U krenelu svaka nit računa novu vrednost samo jednog elementa matrice. Na kraju se vrši kopiranje podatak sa uređaja na host.

2.3. Rezultati

Ulazni parametri	Ubrzanje
30 30 1000	$\frac{52.596287}{10.735872}$ $= 4.899117$
500 500 10	$\frac{139.224091}{1.259296}$ $= 110.557083$
1000 1000 100	$\frac{5609.303223}{19.767649}$ $= 283.761780$
1000 1000 1000	$\frac{55344.406250}{196.816544}$ $= 281.197937$
Srednje ubrzanje	170.10

2.3.1. Logovi izvršavanja

```
width=30, height=30, iteration=1000
Time elapsed, sequential in ms: 52.596287
Time elapsed, parallel in ms: 10.735872
Test PASSED

width=500, height=500, iteration=10
```

```

Time elapsed, sequential in ms: 139.224091
Time elapsed, parallel in ms: 1.259296
Test PASSED

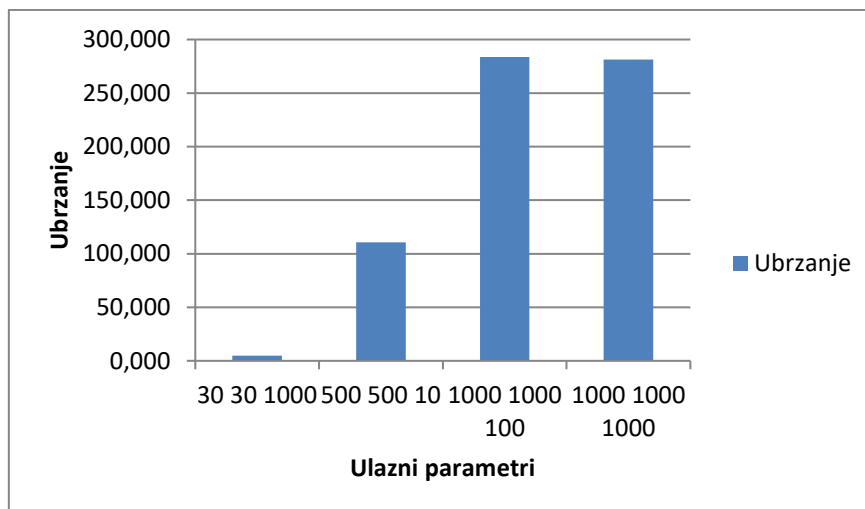
width=1000, height=1000, iteration=100
Time elapsed, sequential in ms: 5609.303223
Time elapsed, parallel in ms: 19.767649
Test PASSED

width=1000, height=1000, iteration=1000
Time elapsed, sequential in ms: 55344.406250
Time elapsed, parallel in ms: 196.816544
Test PASSED

```

Listing 2. Sekvencijalna i paralelna izvršavanja

2.3.2. Grafici ubrzanja



Slika 2. Grafik ubrzanja za različite ulazne parametre

2.3.3. Diskusija dobijenih rezultata

Primećujemo velika ubrzanja u rasponu od 5 puta (za test primer sa malim dimenzijama matrice) do čak 280 puta za veće matrice. Ova ubrzanja su dosta veća u odnosu na OpenMP i MPI biblioteke.

3.PROBLEM 3 – HOTSPOT

3.1. Tekst problema

Paralelizovati program koji rešava problem promene temperature na čipu procesora u dvodimenzionalnom prostoru kroz vreme, ako su poznati početna temperatura i granični uslovi. Simulacija rešava seriju diferencijalnih jednačina nad pravilnom mrežom tačaka kojom se aproksimira površina procesora. Svaka tačka u mreži predstavlja prosečnu temperaturu za odgovarajuću površinu na čipu. Mreža tačaka je predstavljena odgovarajućom matricom koja opisuje trenutne temperature. Program se nalazi u direktorijumu hotspot u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih je od interesa datoteka hotspot.c. Verifikaciju paralelizovanog rešenja vršiti nad dobijenim temperaturama u poslednjem stanju sistema. Koristiti 2D organizaciju jezgra, ukoliko je moguće. Način pokretanja programa se nalazi u datoteci run. Kao pomoćno sredstvo, data je i python skripta koja izlaznu datoteku formatira u heatmap sliku u PNG formatu.

3.2. Delovi koje treba paralelizovati

3.2.1. Diskusija

Diskusija Paralelizacija je izvršena nad nad funkcijom `single_iteration` koju smo podelili na tri funkcije. I svaka od funkcija je obrađivala jedan od tri slučaja (ugao, ivica, sredina).

3.2.2. Način paralelizacije

Prvo je alocirana memorija na uredjaju za potrebne matrice. Kako smo posao podelili na tri funkcije za svaku od funkcija smo odredili koliko blokova niti je potrebno i pozvali smo potrebne kernele. U kernelu svaka nit računa novu vrednost samo jednog elementa matrice. Na kraju se vrši kopiranje podatak sa uredjaja na host.

3.3. Rezultati

Ulaz	Ubrzanja
32 32 8192 1 ./data/temp32_32 ./data/power32_32 ./output/out32_32	$\frac{92.896797}{171.837860}$ = 0.540607
256 256 8192 1 ./data/temp256_256 ./data/power256_256 ./output/out256_256	$\frac{5790.399414}{352.435181}$ = 16.429686
1024 1024 4096 1 ./data/temp1024_1024 ./data/power1024_1024 ./output/out1024_1024_4096	$\frac{46810.578125}{665.509705}$ = 70.337936
1024 1024 8192 1 ./data/temp1024_1024 ./data/power1024_1024 ./output/out1024_1024_8192	$\frac{97473.789062}{1296.680908}$ = 75.171761
1024 1024 16384 1 ./data/temp1024_1024 ./data/power1024_1024 ./output/out1024_1024_16384	$\frac{196322.156250}{2535.055664}$ = 77.442940
1024 1024 32768 1 ./data/temp1024_1024 ./data/power1024_1024 ./output/out1024_1024_32768	$\frac{392788.468750}{5055.701172}$ = 77.692184
Srednje ubrzanje	52.94

3.3.1. Logovi izvršavanja

```
grid_rows=32,      grid_cols=32,      sim_time=8192,      tempfile=./data/temp32_32,
powerfile=./data/power32_32
Time elapsed, sequential in ms: 92.896797
Time elapsed, parallel in ms: 171.837860
Test PASSED

grid_rows=256,      grid_cols=256,      sim_time=8192,      tempfile=./data/temp256_256,
powerfile=./data/power256_256
Time elapsed, sequential in ms: 5790.399414
Time elapsed, parallel in ms: 352.435181
Test PASSED

grid_rows=1024,      grid_cols=1024,      sim_time=4096,      tempfile=./data/temp1024_1024,
powerfile=./data/power1024_1024
Time elapsed, sequential in ms: 46810.578125
Time elapsed, parallel in ms: 665.509705
Test PASSED

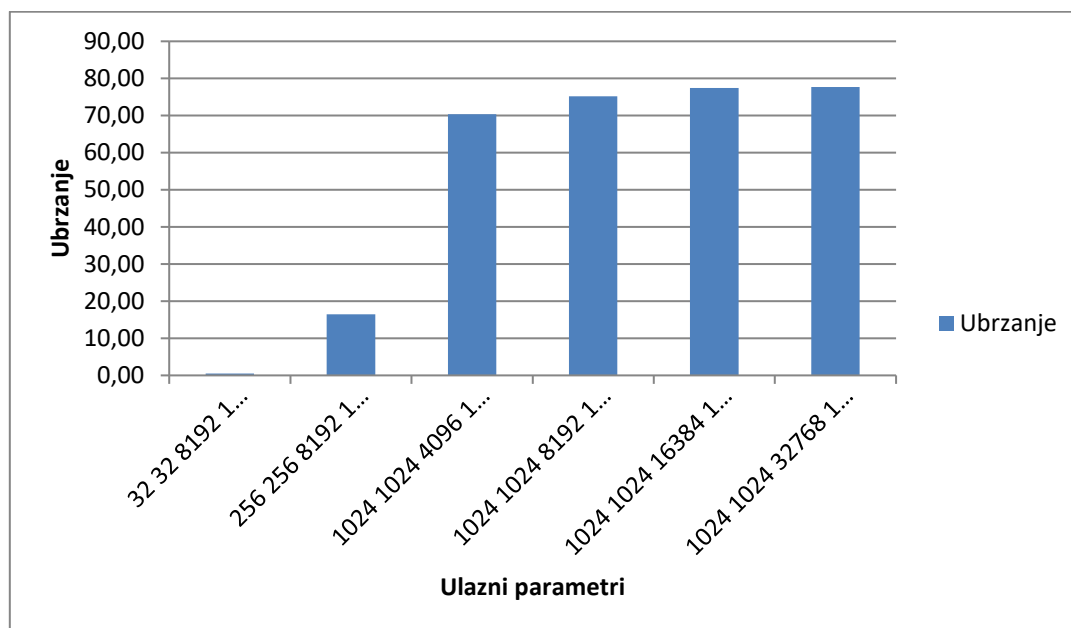
grid_rows=1024,      grid_cols=1024,      sim_time=8192,      tempfile=./data/temp1024_1024,
powerfile=./data/power1024_1024
Time elapsed, sequential in ms: 97473.789062
Time elapsed, parallel in ms: 1296.680908
Test PASSED

grid_rows=1024,      grid_cols=1024,      sim_time=16384,      tempfile=./data/temp1024_1024,
powerfile=./data/power1024_1024
Time elapsed, sequential in ms: 196322.156250
Time elapsed, parallel in ms: 2535.055664
Test PASSED

grid_rows=1024,      grid_cols=1024,      sim_time=32768,      tempfile=./data/temp1024_1024,
powerfile=./data/power1024_1024
Time elapsed, sequential in ms: 392788.468750
Time elapsed, parallel in ms: 5055.701172
Test PASSED
```

Listing 3. Sekvencijalna i paralelna izvršavanja

3.3.2. Grafici ubrzanja



Slika 3. Grafik ubrzanja za različite ulazne parametre

3.3.3. Diskusija dobijenih rezultata

Primećujemo velika ubrzanja za sve test primere osim prvog gde je dimenzija čipa dosta mala. Za ostale ulazne parametre ubrzanje je primetno i u rasponu su od 16 do 78. Ova ubrzanja su dosta veća u odnosu na OpenMP i MPI biblioteke. Usled numeričke nestabilnosti u radu sa realnim podacima imamo mala odstupanja koja su nešto veća od ACCURACY zbog neslaganja na centralnom i grafičkom procesoru.