UNIVERZITET U BEOGRADU - ELEKTROTEHNIČKI FAKULTET

MULTIPROCESORKI SISTEMI (13S114MUPS, 13E114MUPS)

DOMAĆI ZADATAK 4 – CUDA

Izveštaj o urađenom domaćem zadatku

Predmetni asistent:	Studenti:
doc. dr Marko Mišić	Mara Bolić 2017/0614 Edvin Maid 2017/0117

Beograd, novembar 2020.

SADRŽAJ

1. Problem 1 – Pi Calc

1.1. Tekst problema

Paralelizovati program koji izračunava vrednost broja PI korišćenjem formule: $\pi = 4 * \sum (-1)^{k+1} \frac{1}{2k-1}$ n $k=1$. Tačnost izračunavanja direktno zavisi od broja iteracija, a zbog malog radijusa konvergencije serija konvergira veoma sporo. Program se nalazi u datoteci piCalc.c u arhivi koja je priložena uz ovaj dokument. Program testirati sa parametrima koji su dati u datoteci run

1.1.1. Diskusija

U ovom zadatku paralelizovana je jedina *for* petlja. Ostatak koda je sekvencijalan i nema potrebe za njegovom paralelizacijom.

1.1.2. Način paralelizacije

Kompletno izracunavanje broja pi je smešteno u funkciju krenela. Za broj blokova je uzeta gornje zaokružena vrednost količnika broja tačaka i maksimalnog broja niti u bloku odnosno 1024. Taj broj je takodje prosleđen za broj niti.
Nakon jedne iteracije izračunavanja broja pi, svaka nit svoj rezultat smešta u deljenu memoriju i zatim čeka u sinhornizacionoj tački da ostale niti završe svoj deo posla. Na kraju se vrši redukcija vrednosti iz deljene memorije.

1.2. Rezultati

1.2.1. Logovi izvršavanja

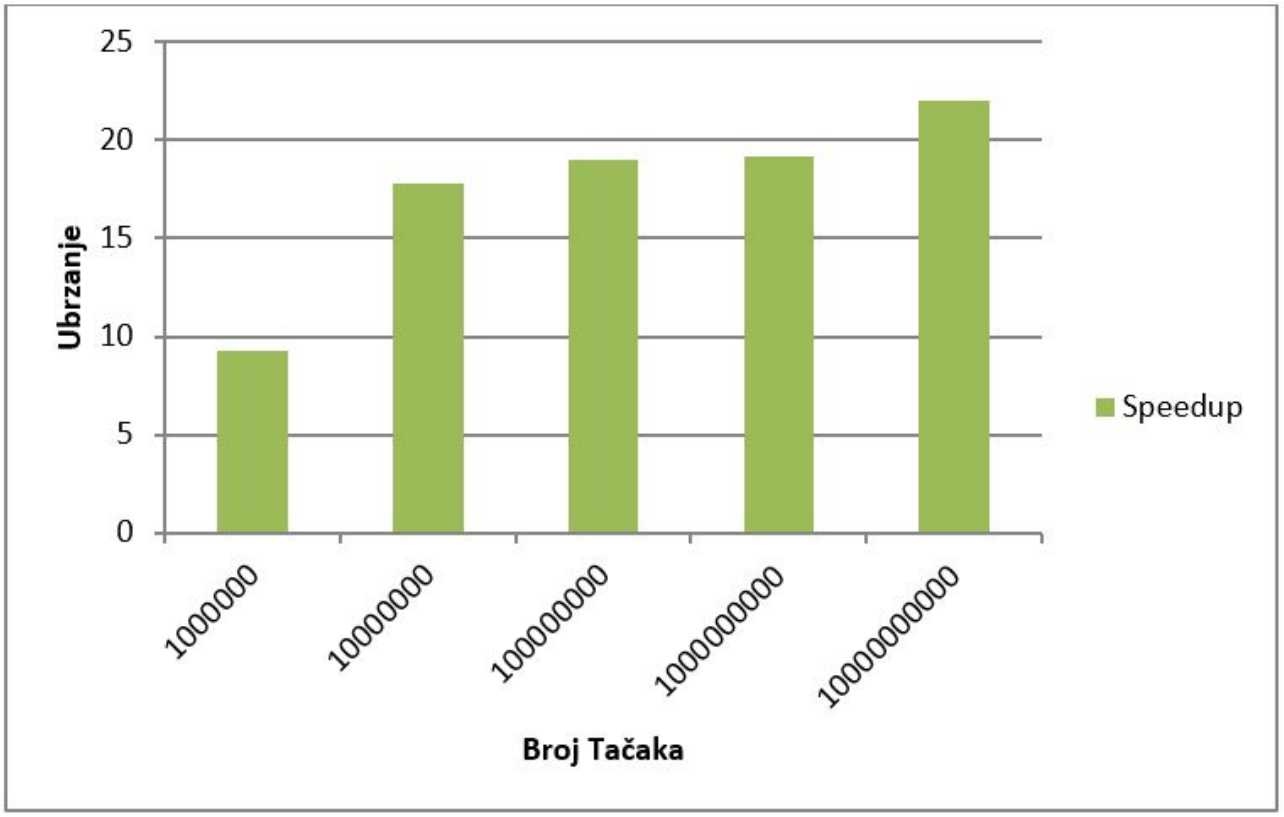
Ovde su dati logovi izvršavanja paralelnog i sekvencijalnog programa za definisane test primere iz *run* fajla.

```
SEQUENTIAL:
With n = 1000000 terms
   Our estimate of pi = 3.14159165358977
Sequential Time: 3.697216
PARALLEL:
   Our estimate of pi = 3.14159165301346
Parallel Time: 0.400000
TEST PASSED
```

```
SEQUENTIAL:
With n = 10000000 terms
   Our estimate of pi = 3.14159255358979
Sequential Time: 31.847744
PARALLEL:
   Our estimate of pi = 3.14159255358342
Parallel Time: 1.787616
TEST PASSED

SEQUENTIAL:
With n = 100000000 terms
   Our estimate of pi = 3.14159264358933
Sequential Time: 318.338959
PARALLEL:
   Our estimate of pi = 3.14159264358983
Parallel Time: 16.778145
TEST PASSED

SEQUENTIAL:
With n = 1000000000 terms
   Our estimate of pi = 3.14159265258805
Sequential Time: 3182.947266
PARALLEL:
   Our estimate of pi = 3.14159265258984
Parallel Time: 166.006363
TEST PASSED

SEQUENTIAL:
With n = 10000000000 terms
   Our estimate of pi = 3.14159265348835
Sequential Time: 31832.091797
PARALLEL:
   Our estimate of pi = 3.14159265319150
Parallel Time: 1445.960815
TEST PASSED
```

Listing 1. Sekvencijalna i paralelna izvršavanja Pi Calc

1.2.2. Grafici ubrzanja



Grafik zavisnosti ubrzanja od broja tačaka

1.2.3. Diskusija dobijenih rezultata

Za veliki broj tačaka se dobija oko 20 puta veća brzina dok za mali broj tačaka dostiže ubrzanje malo manje od 10 puta.

Problem 2 – Needleman-Wunsch

1.1. Tekst problema

Paralelizovati program koji vrši poravnavanje bioloških sekvenci korišćenjem Needleman-Wunsch algoritma. Algoritam predstavlja primenu koncepta dinamičkog programiranja za globalno poravnavanje dve sekvence nukleotida ili aminokiselina (više o algoritmu na adresi: https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm). Program se nalazi u datoteci needle.c u arhivi koja je priložena uz ovaj dokument. Obratiti pažnju na mogućnost korišćenja deljene memorije radi ubrzavanja pristupa podacima. Koristiti 2D organizaciju jezgra, ukoliko je moguće. Program testirati sa parametrima koji su dati u datoteci run

1.1.1. Diskusija

Ovaj problem ima 2 krucijalne petlje koje se mogu paralelizovati i obavljaju glavni deo računanja za ovaj algoritam. Koje se medjusobno zavisne i moraju da se izračunaju uzastopno.

1.1.2. Način paralelizacije

Izračunavanje gornje leve polovine matrice je smesteno u jednu a izračunavanje donje desne matrice u drugu funkciju kernela. Obe funkcije kernlena su pozvane za svaku dijaguonalu sa odgovarajucim brojem blokova koji svi poseduju po 1024 niti. Ovo se radi jer su dijagonale medjusobno zavisne pa je da bi se odrzao poredak racunanja zbog korektnog resenja svaka dijagonala se racuna u zasebnom kernelu.

1.2. Rezultati

1.2.1. Logovi izvršavanja

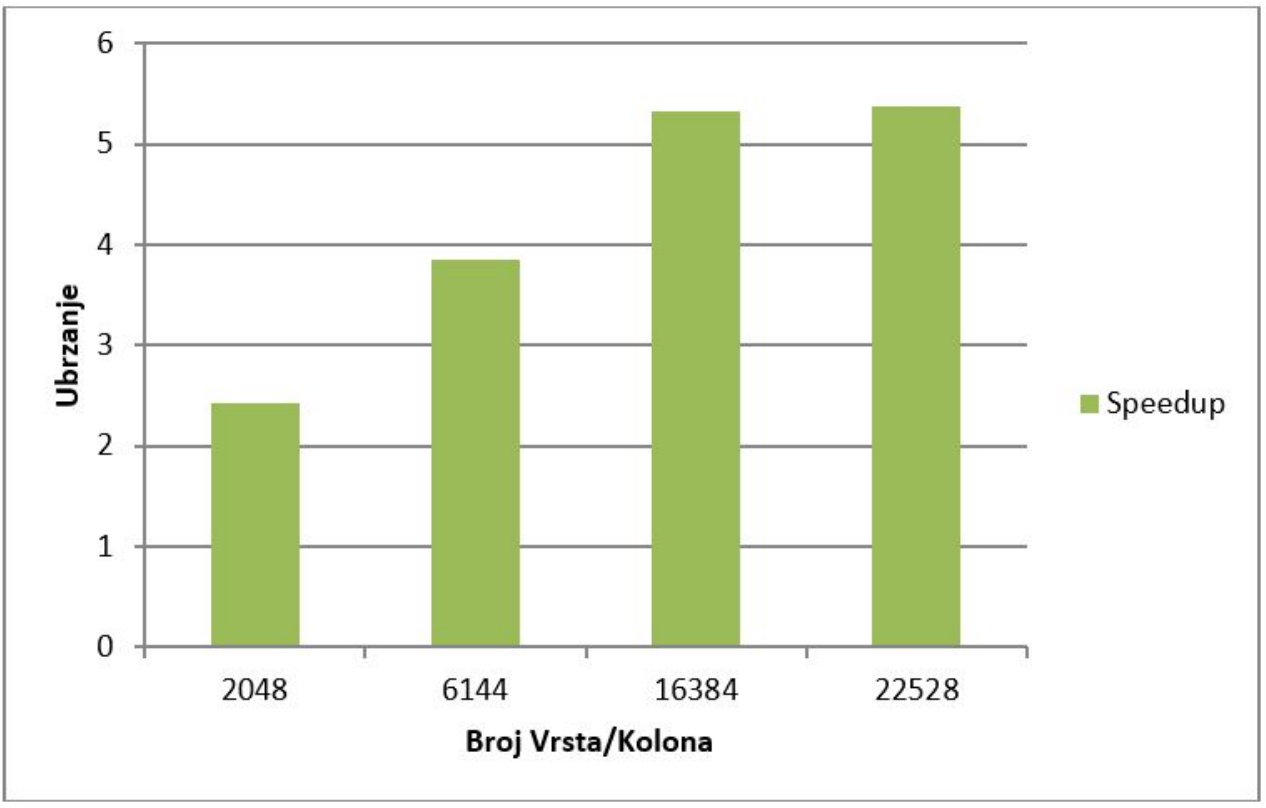
Ovde su dati logovi izvršavanja paralelnog i sekvencijalnog programa za definisane test primere iz *run* fajla.

```
 2048 10
SEQUENTIAL
The process took: 212.134720
PARALLEL
The process took: 87.695229
TEST PASSED

 6144 10
SEQUENTIAL
The process took: 1892.403198
PARALLEL
The process took: 491.558380
TEST PASSED

 16384 10
SEQUENTIAL
The process took: 14526.247070
PARALLEL
The process took: 2727.286377
TEST PASSED

 22528 10
SEQUENTIAL
The process took: 28265.501953
PARALLEL
The process took: 5264.120605
TEST PASSED
```

Listing 1. Sekvencijalno izvršavanje NEEDLEMAN-WUNSCH N=2

1.2.2. Grafici ubrzanja

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



Grafik zavisnosti ubrzanja u zavisnosti od dimenzija matrice

1.2.3. Diskusija dobijenih rezultata

Sa povecanjem dimenzija problema se povecava i ubrzanje koje se dostize i za 22528 se dostize vrhunac ubrzanja naseg resenja.

Problem 3 – Nbody

1.3. Tekst problema

Paralelizovati program koji simulira problem interakcije čvrstih tela u dvodimenzionalnom prostoru (nbody problem). Tela interaguju putem gravitacione sile na osnovu sopstvene mase, pozicije u prostoru i trenutne brzine. Program se nalazi u direktorijumu nbody u arhivi koja je priložena uz ovaj dokument. Program se sastoji od više datoteka, od kojih je od interesa datoteka nbody.c. Analizirati dati kod i obratiti pažnju na način izračunavanja sila i energija. Obratiti pažnju na efikasnost paralelizacije, mogućnost upotrebe deljene memorije i potrebu za redukcijom. Verifikaciju paralelizovanog rešenja vršiti nad dobijenim energijama i poslednjem stanju sistema. Način pokretanja programa se nalazi u datoteci run.

1.3.1. Diskusija

Petlja za računaje kinetičke energije u Compute_energy funkciji se slobodno paralelizuje. Može se paralelizovati ili unutrašnja (ugnježdena) ili spoljašnja petlja za računanje potencijalne energije. U ovom slučaju je jednostavnije vršiti paralelizaciju spoljašnje petlje.

1.3.2. Način paralelizacije

Na počeku smo pokušali da u *Compute_energy* i *Compute_forces* niz *current* smestimo u deljenu memoriju kako bi se smanjio brze dohvatali elementi koji su često korišćeni. Mađutim, ovaj metod nije doveo do kvalitetnih ubrzanja za ulazne podatke. Ali nakon sto smo prešli na trenutni način paralelizacije dobijena tačna rešenja i primetno veća brzina.

Ovde smo for petlju podelili na nezavisno pozivanje iteracija petlji. U *Compute_energy* kernelu se za svaku česticu potpuno odvojeno po nitima racuna energija. Nakon ovih izračunavanja se pokreće postupak redukcije energija svih čestica da dobijemo ukupne energije.

Paralelizacija *Compute_forces* i *Update_part* funkcija je vršena tako što je svakoj čestici bila dodeljena nit koja računa silu koja deluje na tu česticu tako što računa sve moguće parove i njihove reakcije.

1.4. Rezultati

1.4.1. Logovi izvršavanja

Ovde su dati logovi izvršavanja paralelnog i sekvencijalnog programa za definisane test primere iz *run* fajla.

```
 100
SEQUENTIAL
   PE = -6.985593e+36, KE = 2.250000e+35, Total Energy = -6.760593e+36
   PE = -7.035612e+36, KE = 1.304554e+36, Total Energy = -5.731058e+36
```

```
The process took: 42.991489
PARALLEL
   PE = -6.985593e+36, KE = 2.250000e+35, Total Energy = -6.760593e+36
   PE = -7.035612e+36, KE = 1.304554e+36, Total Energy = -5.731058e+36
The process took: 79.932510
Test PASSED



 500
SEQUENTIAL
   PE = -4.831939e+37, KE = 1.125000e+36, Total Energy = -4.719439e+37
   PE = -4.754056e+37, KE = 1.360414e+36, Total Energy = -4.618014e+37
The process took: 1067.382202
PARALLEL
   PE = -4.831939e+37, KE = 1.125000e+36, Total Energy = -4.719439e+37
   PE = -4.754056e+37, KE = 1.360414e+36, Total Energy = -4.618014e+37
The process took: 491.654327
Test PASSED

5000
SEQUENTIAL
   PE = -6.751832e+38, KE = 1.125000e+37, Total Energy = -6.639332e+38
   PE = -6.649074e+38, KE = 2.481116e+36, Total Energy = -6.624263e+38
The process took: 106567.390625
PARALLEL
   PE = -6.751832e+38, KE = 1.125000e+37, Total Energy = -6.639332e+38
   PE = -6.649074e+38, KE = 2.481116e+36, Total Energy = -6.624263e+38
The process took: 7714.484375
Test PASSED
```

1.4.2. Grafici ubrzanja

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

Grafik zavisnosti ubrzanja u zavisnosti od broja čestica

1.4.3. Diskusija dobijenih rezultata

Za test primer od 100 čestica se dobija usporenje zbog velikih overheada. Već se za 500 čestica primecuje ubrzanje oko 2 puta dok se za najveći broj čestica dobija do čak 14 puta veća brzina.