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 | Lazar Cvetković 2016/0127  Strahinja Stefanović 2016/0130 |

Beograd, maj 2020.

Sadržaj

[Sadržaj 2](#_Toc38566844)

[1. Problem 1 – Julia Set (podela posla) 3](#_Toc38566845)

[1.1. Tekst problema 3](#_Toc38566846)

[1.2. Delovi koje treba paralelizovati 3](#_Toc38566847)

[1.2.1. Diskusija 3](#_Toc38566848)

[1.2.2. Način paralelizacije 3](#_Toc38566849)

[1.3. Rezultati 3](#_Toc38566850)

[1.3.1. Logovi izvršavanja 3](#_Toc38566851)

[1.3.2. Grafici ubrzanja 9](#_Toc38566852)

[1.3.3. Diskusija dobijenih rezultata 9](#_Toc38566853)

[2. Problem 2 – Julia Set (manager - worker) 10](#_Toc38566854)

[2.1. Tekst problema 10](#_Toc38566855)

[2.2. Delovi koje treba paralelizovati 10](#_Toc38566856)

[2.2.1. Diskusija 10](#_Toc38566857)

[2.2.2. Način paralelizacije 10](#_Toc38566858)

[2.3. Rezultati 10](#_Toc38566859)

[2.3.1. Logovi izvršavanja 10](#_Toc38566860)

[2.3.2. Grafici ubrzanja 16](#_Toc38566861)

[2.3.3. Diskusija dobijenih rezultata 16](#_Toc38566862)

[3. Problem 3 – Izoštravanje slike 17](#_Toc38566863)

[3.1. Tekst problema 17](#_Toc38566864)

[3.2. Delovi koje treba paralelizovati 17](#_Toc38566865)

[3.2.1. Diskusija 17](#_Toc38566866)

[3.2.2. Način paralelizacije 17](#_Toc38566867)

[3.3. Rezultati 17](#_Toc38566868)

[3.3.1. Logovi izvršavanja 17](#_Toc38566869)

[3.3.2. Grafici ubrzanja 21](#_Toc38566870)

[3.3.3. Diskusija dobijenih rezultata 21](#_Toc38566871)

[4. Problem 4 – MRI Gridding 22](#_Toc38566872)

[4.1. Tekst problema 22](#_Toc38566873)

[4.2. Delovi koje treba paralelizovati 22](#_Toc38566874)

[4.2.1. Diskusija 22](#_Toc38566875)

[4.2.2. Način paralelizacije 22](#_Toc38566876)

[4.3. Rezultati 22](#_Toc38566877)

[4.3.1. Logovi izvršavanja 22](#_Toc38566878)

[4.3.2. Grafici ubrzanja 24](#_Toc38566879)

[4.3.3. Diskusija dobijenih rezultata 24](#_Toc38566880)

1. Problem 1 – Julia Set (podela posla)
   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. Proces sa rangom 0 treba da učita ulazne podatke, raspodeli posao ostalim procesima, na kraju prikupi dobijene rezultate i ravnopravno učestvuje u obradi. Za razmenu podataka, koristiti rutine za kolektivnu komunikaciju. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

* 1. Delovi koje treba paralelizovati
     1. Diskusija

Moguće je paralelizovati samo glavnu funkciju koja pokreće izračunavanje svih tačaka skupa **julia\_set**, odnosno dvostruku for petlju unutar nje. Funkciju **julia** koja izračunava tačnu vrednost tačke nije moguće paralelizovati zato što postoji zavisnost između susednih iteracija petlje koju ona izvršava.

* + 1. Način paralelizacije

Paralelizacija je izvršena jednakom podelom poslova među MPI procesima.

* 1. Rezultati
     1. Logovi izvršavanja

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

W: 500, H: 500

Sequential execution time: 0.038702

Parallel execution time: 0.000205

Speedup: 188.679800

Test PASSED

--------------------

W: 500, H: 500

Sequential execution time: 0.045815

Parallel execution time: 0.000337

Speedup: 136.056142

Test PASSED

--------------------

W: 500, H: 500

Sequential execution time: 0.048258

Parallel execution time: 0.000430

Speedup: 112.340772

Test PASSED

--------------------

W: 1000, H: 1000

Sequential execution time: 0.147114

Parallel execution time: 0.000503

Speedup: 292.394116

Test PASSED

--------------------

W: 1000, H: 1000

Sequential execution time: 0.182975

Parallel execution time: 0.000882

Speedup: 207.435829

Test PASSED

--------------------

W: 1000, H: 1000

Sequential execution time: 0.188390

Parallel execution time: 0.001158

Speedup: 162.633983

Test PASSED

--------------------

W: 1000, H: 2000

Sequential execution time: 0.294940

Parallel execution time: 0.000861

Speedup: 342.634747

Test PASSED

--------------------

W: 1000, H: 2000

Sequential execution time: 0.365545

Parallel execution time: 0.001552

Speedup: 235.550997

Test PASSED

--------------------

W: 1000, H: 2000

Sequential execution time: 0.378153

Parallel execution time: 0.002012

Speedup: 187.969737

Test PASSED

--------------------

Listing 1. Izlaz programa

* + 1. Grafici ubrzanja

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

Slika 1. Grafik zavisnosti ubrzanja

* + 1. Diskusija dobijenih rezultata

Primetno je da MPI sa jednim procesom radi sporije, zbog režijskih troškova, dok se postiže zadovoljavajuće ubrzanje pri 2 i 4 procesa. Kada je broj procesa 8, tada očigledno dolazi do prevelikog zagušenja/zasićenja resursa mašine.

1. Problem 2 – Izoštravanje slike

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

* 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. Ukoliko je moguće, koristiti rutine za neblokirajuću komunikaciju za razmenu poruka. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

* 1. Delovi koje treba paralelizovati
     1. Diskusija

Izvršena je paralelizacija četiri ugnježdene petlje koje izračunavaju konvoluciju, kao i sledeće dvostruke petlje koja taj dobijeni rezultat koristi za izoštravanje slike.

* + 1. Način paralelizacije

Korišćena je podela poslova, td. svaki MPI proces dobije podjednak deo za izvršavanje. Pri računanju puno vremena se trošilo na aritmetičkim operacijama nad filterMatrix, te je stoga ona implementirana kao LUT.

* 1. Rezultati
     1. Logovi izvršavanja

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

Input file: data\_dz4z2/balloons\_noisy.pgm

Sequential execution time: 2.771515

Parallel execution time: 0.035567

Speedup: 77.922778

Test PASSED

Input file: data\_dz4z2/bone\_scint.pgm

Sequential execution time: 18.779737

Parallel execution time: 0.188444

Speedup: 99.657030

Test PASSED

Input file: data\_dz4z2/fuzzy.pgm

Sequential execution time: 3.887741

Parallel execution time: 0.046745

Speedup: 83.169148

Test PASSED

Input file: data\_dz4z2/lena512.pgm

Sequential execution time: 2.365053

Parallel execution time: 0.025276

Speedup: 93.569826

Test PASSED

Input file: data\_dz4z2/man.pgm

Sequential execution time: 9.362493

Parallel execution time: 0.099027

Speedup: 94.545179

Test PASSED

Input file: data\_dz4z2/Rainier\_blur.pgm

Sequential execution time: 18.249267

Parallel execution time: 0.184669

Speedup: 98.821759

Test PASSED

Listing 3. Ispis programa

* + 1. Grafici ubrzanja

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

Slika 3. Grafik zavisnosti ubrzanja

* + 1. Diskusija dobijenih rezultata

Primetno je značajno ubrzanje izvršavanja, i do 20-25 puta. Pritom, ona raste sa povećanjem broja procesa, gde ulazi u zasićenje pri slučaju sa osam procesa. Pri računanju puno vremena se trošilo na aritmetičkim operacijama nad filterMatrix, te je stoga ona implementirana kao LUT.

1. Problem 4 – MRI Gridding

U okviru ovog poglavlja je dat kratak izveštaj u vezi rešenja zadatog problema 4.

* 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, kao i na različite mogućnosti i nivoe na kojima se može obaviti paralelizacija koda. 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]

* 1. Delovi koje treba paralelizovati
     1. Diskusija

Paralelizovana je funkcija **gridding\_Gold** koju ima i najviše smisla paralelizovati pošto su u njoj izvršava sav potreban račun.

* + 1. Način paralelizacije

Paralelizacija je rađena metodom podele poslova, tako što je spoljna petnja podeljena na jednake delove koje uzima svaki proces, a onda se vršila redukcija nad gridData i sampleDensity, nad kojim su bili definisani nestandardni tipovi i specifičan operator redukcije.

* 1. Rezultati
     1. Logovi izvršavanja

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

Number of threads: 1

Sequential execution time: 4.447496

Parallel execution time: 4.503017

TEST PASSED - gridData

TEST PASSED - sampleDensity

Number of threads: 2

Sequential execution time: 4.477351

Parallel execution time: 2.477347

TEST PASSED - gridData

TEST PASSED - sampleDensity

Number of threads: 4

Sequential execution time: 4.455474

Parallel execution time: 1.481890

TEST PASSED - gridData

TEST PASSED - sampleDensity

Number of threads: 8

Sequential execution time: 4.483704

Parallel execution time: 1.270220

TEST PASSED - gridData

TEST PASSED - sampleDensity

Listing 4. Ispis programa

* + 1. Grafici ubrzanja

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

Slika 5. Grafik zavisnosti ubrzanja

* + 1. Diskusija dobijenih rezultata

Režijski troškovi MPI biblioteke dolaze do izražaja pri postavci od jednog procesa, inače se postiže očekivano i značajno ubrzanje i do četiri puta u odnosu na sekvencijalno izvršavanje.