Univerzitet u Beogradu - Elektrotehnički fakultet

Multiprocesorki sistemi (13S114MUPS, 13E114MUPS)



Domaći zadatak 1 – OPENMP

Izveštaj o urađenom domaćem zadatku

|  |  |
| --- | --- |
| Predmetni saradnici: | Studenti: |
| doc. dr Marko Mišić  dipl. ing. Pavle Divović | Andrea Popović 2018/0316  Vukašin Nedeljković 2018/0217 |

Beograd, april 2022.

Sadržaj

[Sadržaj 2](#_Toc100409231)

[1. Problem 1 - SIMPLEX 4](#_Toc100409232)

[1.1. Tekst problema 4](#_Toc100409233)

[1.2. Delovi koje treba paralelizovati 4](#_Toc100409234)

[1.2.1. Diskusija 4](#_Toc100409235)

[1.2.2. Način paralelizacije 4](#_Toc100409236)

[1.3. Rezultati 5](#_Toc100409237)

[1.3.1. Logovi izvršavanja 5](#_Toc100409238)

[1.3.2. Grafici ubrzanja 5](#_Toc100409239)

[1.3.3. Diskusija dobijenih rezultata 7](#_Toc100409240)

[2. PROBLEM 2 – SIMPLEX 8](#_Toc100409241)

[2.1. Tekst problema 8](#_Toc100409242)

[2.2. Delovi koje treba paralelizovati 8](#_Toc100409243)

[2.2.1. Diskusija 8](#_Toc100409244)

[2.2.2. Način paralelizacije 8](#_Toc100409245)

[2.3. Rezultati 9](#_Toc100409246)

[2.3.1. Logovi izvršavanja 9](#_Toc100409247)

[2.3.2. Grafici ubrzanja 11](#_Toc100409248)

[2.3.3. Diskusija dobijenih rezultata 12](#_Toc100409249)

[3. PROBLEM 3 – GAME OF LIFE 13](#_Toc100409250)

[3.1. Tekst problema 13](#_Toc100409251)

[3.2. Delovi koje treba paralelizovati 13](#_Toc100409252)

[3.2.1. Diskusija 13](#_Toc100409253)

[3.2.2. Način paralelizacije 14](#_Toc100409254)

[3.3. Rezultati 14](#_Toc100409255)

[3.3.1. Logovi izvršavanja 14](#_Toc100409256)

[3.3.2. Grafici ubrzanja 17](#_Toc100409257)

[3.3.3. Diskusija dobijenih rezultata 17](#_Toc100409258)

[4. PROBLEM 4 – GAME OF LIFE 18](#_Toc100409259)

[4.1. Tekst problema 18](#_Toc100409260)

[4.2. Delovi koje treba paralelizovati 18](#_Toc100409261)

[4.2.1. Diskusija 18](#_Toc100409262)

[4.2.2. Način paralelizacije 18](#_Toc100409263)

[4.3. Rezultati 19](#_Toc100409264)

[4.3.1. Logovi izvršavanja 19](#_Toc100409265)

[4.3.2. Grafici ubrzanja 22](#_Toc100409266)

[4.3.3. Diskusija dobijenih rezultata 22](#_Toc100409267)

[5. PROBLEM 5 – HOTSPOT 24](#_Toc100409268)

[5.1. Tekst problema 24](#_Toc100409269)

[5.2. Delovi koje treba paralelizovati 24](#_Toc100409270)

[5.2.1. Diskusija 24](#_Toc100409271)

[5.2.2. Način paralelizacije 24](#_Toc100409272)

[5.3. Rezultati 25](#_Toc100409273)

[5.3.1. Logovi izvršavanja 25](#_Toc100409274)

[5.3.2. Grafici ubrzanja 29](#_Toc100409275)

[5.3.3. Diskusija dobijenih rezultata 30](#_Toc100409276)

1. Problem 1 - SIMPLEX
   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 naalazi u datoteci simplex.c. U izvornom kodu data je matrica eksponenata jednačine i ivica simplexa. Ulazni parametar programa je broj iteracija aproksimacije. 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. Delovi koje treba paralelizovati
     1. Diskusija

Uočili smo par potencijalnih funkcija za paralelizaciju. Nakon njihove paralelizacije smo zaključili da nam najbolje performanse daje paralelizacija for petlje u funkciji run. Ostale funkcije ne vredi paralelizovati jer je granularnost prevelika.

* + 1. Način paralelizacije

Zbog nemogućnosti korišćenja worksharing direktiva, odlučili smo da paralelizaciju izvršimo pomoću paralelnog regiona, dok smo posao podelili pomoću promenljivih start, end, chunk, myid, nthreads, gde je svaka nit dobila jednak deo posla.

* 1. Rezultati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| broj niti  ulaz | 1 | 2 | 4 | 8 |
| 500000 | = 1.00 | = 1.86 | = 3.17 | = 4.11 |
| 100000 | = 1.00 | = 1.85 | = 3.18 | = 3.94 |
| 1000000 | = 1.00 | = 1.84 | = 3.08 | = 3.91 |
| **Srednje ubrzanje** | **1.00** | **1.85** | **3.14** | **3.99** |

* + 1. Logovi izvršavanja

Simplex 50000

Time elapsed in ms (sequential program): 870.309042

Time elapsed in ms (parallel program): 868.068687

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1826.704564

Time elapsed in ms (parallel program): 1818.316243

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15764.889866

Time elapsed in ms (parallel program): 15740.025762

Test PASSED

Listing 1. Sekvencijalna i paralelna izvršavanja sa 1 niti

Simplex 50000

Time elapsed in ms (sequential program): 877.010990

Time elapsed in ms (parallel program): 471.146965

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1824.771651

Time elapsed in ms (parallel program): 986.961746

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15786.965556

Time elapsed in ms (parallel program): 8584.325632

Test PASSED

Listing 2. Sekvencijalna i paralelna izvršavanja sa 2 niti

Simplex 50000

Time elapsed in ms (sequential program): 878.761458

Time elapsed in ms (parallel program): 277.447496

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1839.987779

Time elapsed in ms (parallel program): 577.958581

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15909.011163

Time elapsed in ms (parallel program): 5153.458449

Test PASSED

Listing 3. Sekvencijalna i paralelna izvršavanja sa 4 niti

Simplex 50000

Time elapsed in ms (sequential program): 909.308585

Time elapsed in ms (parallel program): 221.371720

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1836.755510

Time elapsed in ms (parallel program): 466.917801

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15658.180819

Time elapsed in ms (parallel program): 4008.137071

Test PASSED

Listing 4. Sekvencijalna i paralelna izvršavanja sa 8 niti

* + 1. Grafici ubrzanja

Slika 1. Grafik zavisnosti ubrzanja od broja niti

* + 1. Diskusija dobijenih rezultata

Na osnovu dobijenih rezultata možemo videti da je postignuto primetno ubrzanje sa paralelizacijom. Takođe sa povećanjem broja niti povećava se i ubrzanje, najbolje ubrzanje je sa 8 niti i to oko 4 puta brže u odnosu na sekvencijalno.

1. PROBLEM 2 – SIMPLEX
   1. Tekst problema

Prethodni program paralelizovati korišćenjem direktiva za podelu posla (worksharing direktive). 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. Delovi koje treba paralelizovati
     1. Diskusija

Uočili smo par potencijalnih funkcija za paralelizaciju. Nakon njihove paralelizacije smo zaključili da nam najbolje performanse daje paralelizacija for petlje u funkciji run. Ostale funkcije ne vredi paralelizovati jer je granularnost prevelika.

* + 1. Način paralelizacije

Paralelizujemo isti deo koda kao u problemu 1 samo sada koristimo worksharing direktivu for kao što je traženo tekstom zadatka. Probali smo različite rasporede schedule i utvrdili smo da je najbolji sa schedule(static).

* 1. Rezultati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| broj niti  ulaz | 1 | 2 | 4 | 8 |
| 50000 | = 1.03 | = 1.86 | = 3.13 | = 3.96 |
| 100000 | = 1.01 | = 1.87 | = 3.16 | = 4.09 |
| 1000000 | = 1.00 | = 1.83 | = 3.04 | = 4.11 |
| **Srednje ubrzanje** | **1.01** | **1.85** | **3.11** | **4.04** |

* + 1. Logovi izvršavanja

Simplex 50000

Time elapsed in ms (sequential program): 897.346314

Time elapsed in ms (parallel program): 873.633010

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1831.686714

Time elapsed in ms (parallel program): 1812.717539

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15715.454876

Time elapsed in ms (parallel program): 15675.815409

Test PASSED

Listing 5. Sekvencijalna i paralelna izvršavanja sa 1 niti

Simplex 50000

Time elapsed in ms (sequential program): 876.437085

Time elapsed in ms (parallel program): 470.557831

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1850.151000

Time elapsed in ms (parallel program): 987.194650

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15663.189551

Time elapsed in ms (parallel program): 8554.116579

Test PASSED

Listing 6. Sekvencijalna i paralelna izvršavanja sa 2 niti

Simplex 50000

Time elapsed in ms (sequential program): 876.851440

Time elapsed in ms (parallel program): 279.833051

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1823.954493

Time elapsed in ms (parallel program): 577.371429

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 15697.520661

Time elapsed in ms (parallel program): 5155.116142

Test PASSED

Listing 7. Sekvencijalna i paralelna izvršavanja sa 4 niti

Simplex 50000

Time elapsed in ms (sequential program): 879.665203

Time elapsed in ms (parallel program): 222.400882

Test PASSED

Simplex 100000

Time elapsed in ms (sequential program): 1837.696114

Time elapsed in ms (parallel program): 449.157040

Test PASSED

Simplex 1000000

Time elapsed in ms (sequential program): 16184.729608

Time elapsed in ms (parallel program): 3937.199700

Test PASSED

Listing 8. Sekvencijalna i paralelna izvršavanja sa 8 niti

* + 1. Grafici ubrzanja

Slika 2. Grafik zavisnosti ubrzanja od broja niti

Slika 3. Grafik zavisnosti ubrzanja od broja niti sa i bez worksharing direktive

* + 1. Diskusija dobijenih rezultata

Na osnovu dobijenih rezultata možemo videti da je postignuto ubrzanje sa paralelizacijom. Takođe sa povećanjem broja niti povećava se i ubrzanje. Takođe možemo primetiti da su ubrzanja skoro identična sa minimalnim razlikama u odnosu na prošli problem gde nismo koristili worksharing direktive.

1. PROBLEM 3 – GAME OF LIFE
   1. Tekst problema

Paralelizovati program koji implementira simulaciju ćelijskog automata Game of Life. Simulacija je predstavljena dvodimenzionalnom matricom dimenzija w x 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. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

* 1. Delovi koje treba paralelizovati
     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 ugneždena petlja ima mali broj iteracija i u njoj je samo provera uslova pa se ni nju ne isplati optimizovati. Zbog toga paralelizujemo prvu ugneždenu petlju.

* + 1. Način paralelizacije

Korišćena je worksharing direktiva for sa odredbom collapse(1). Probali smo različite rasporede schedule i utvrdili smo da je najbolji sa schedule(stataic, 5).

* 1. Rezultati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| broj niti  ulazni parametri | 1 | 2 | 4 | 8 |
| 30 30 1000 | = 1.14 | = 2.10 | = 3.17 | = 3.67 |
| 500 500 10 | = 0.94 | = 1.90 | = 3.56 | = 4.03 |
| 1000 1000 100 | = 0.93 | = 1.83 | = 3.38 | = 4.48 |
| 1000 1000 1000 | = 0.93 | = 1.84 | = 3.73 | = 4.36 |
| **Srednje ubrzanje** | **0.99** | **1.92** | **3.46** | **4.14** |

* + 1. Logovi izvršavanja

width=30, height=30, iteration=1000

Time elapsed, sequential in ms: 33.947407

Time elapsed, parallel in ms: 29.777111

Test PASSED

width=500, height=500, iteration=10

Time elapsed, sequential in ms: 72.104996

Time elapsed, parallel in ms: 76.487408

Test PASSED

width=1000, height=1000, iteration=100

Time elapsed, sequential in ms: 2896.550021

Time elapsed, parallel in ms: 3113.850245

Test PASSED

width=1000, height=1000, iteration=1000

Time elapsed, sequential in ms: 28539.086630

Time elapsed, parallel in ms: 30582.420922

Test PASSED

Listing 9. Sekvencijalna i paralelna izvršavanja sa 1 niti

width=30, height=30, iteration=1000

Time elapsed, sequential in ms: 32.948428

Time elapsed, parallel in ms: 15.704965

Test PASSED

width=500, height=500, iteration=10

Time elapsed, sequential in ms: 73.824149

Time elapsed, parallel in ms: 38.895495

Test PASSED

width=1000, height=1000, iteration=100

Time elapsed, sequential in ms: 2897.931875

Time elapsed, parallel in ms: 1584.834766

Test PASSED

width=1000, height=1000, iteration=1000

Time elapsed, sequential in ms: 29133.896641

Time elapsed, parallel in ms: 15868.059293

Test PASSED

Listing 10. Sekvencijalna i paralelna izvršavanja sa 2 niti

width=30, height=30, iteration=1000

Time elapsed, sequential in ms: 35.391786

Time elapsed, parallel in ms: 11.179693

Test PASSED

width=500, height=500, iteration=10

Time elapsed, sequential in ms: 71.557198

Time elapsed, parallel in ms: 20.091148

Test PASSED

width=1000, height=1000, iteration=100

Time elapsed, sequential in ms: 2897.726081

Time elapsed, parallel in ms: 856.093507

Test PASSED

width=1000, height=1000, iteration=1000

Time elapsed, sequential in ms: 29910.154797

Time elapsed, parallel in ms: 8011.970540

Test PASSED

Listing 11. Sekvencijalna i paralelna izvršavanja sa 4 niti

width=30, height=30, iteration=1000

Time elapsed, sequential in ms: 36.238849

Time elapsed, parallel in ms: 9.876280

Test PASSED

width=500, height=500, iteration=10

Time elapsed, sequential in ms: 70.960523

Time elapsed, parallel in ms: 17.610512

Test PASSED

width=1000, height=1000, iteration=100

Time elapsed, sequential in ms: 2943.773821

Time elapsed, parallel in ms: 661.860891

Test PASSED

width=1000, height=1000, iteration=1000

Time elapsed, sequential in ms: 28589.022624

Time elapsed, parallel in ms: 6553.778025

Test PASSED

Listing 12. Sekvencijalna i paralelna izvršavanja sa 8 niti

* + 1. Grafici ubrzanja

Slika 4. Grafik zavisnosti ubrzanja od broja niti

* + 1. Diskusija dobijenih rezultata

Na osnovu dobijenih rezultata možemo videti da je postignuto ubrzanje sa paralelizacijom. Takođe sa povećanjem broja niti povećava se i ubrzanje. Takođe nije bilo isplativo meriti vreme jednog izvršavanja funkcije evolve jer je ono previše malo, pa je mereno vreme samo na nivou cele simulacije.

1. PROBLEM 4 – GAME OF LIFE
   1. Tekst problema

Rešiti prethodni problem korišćenjem koncepta poslova (tasks). Obratiti pažnju na eventualnu potrebu za sinhronizacijom. Rešenje testirati i prilagoditi tako da granularnost poslova bude optimalna. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

* 1. Delovi koje treba paralelizovati
     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 ugneždena petlja ima mali broj iteracija i u njoj je samo provera uslova pa se ni nju ne isplati optimizovati. Zbog toga paralelizujemo prvu ugneždenu petlju.

* + 1. Način paralelizacije

Tekstom zadatka je zahtevano da se problem reši konceptom poslova(taks). Pošto imamo ugneždenu petlju postoje dve mogućnosti. Prva je da u jedan task pakujemo telo unutrašnje petlje, dok je druga da u jedan task pakujemo celu unutrašnju petlju. Bolje rezultate nam je dala druga mogućnost. Takođe ovde nije bilo potrebe koristiti bilo kakvu sinhronizaciju jer su podaci u tasku međusobno nezavisni, a na kraju single direktive imamo implicitnu barijeru.

* 1. Rezultati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| broj niti  ulazni parametri | 1 | 2 | 4 | 8 |
| 30 30 1000 | = 1.16 | = 1.98 | = 2.31 | = 1.17 |
| 500 500 10 | = 1.04 | = 2.10 | = 3.85 | = 4.43 |
| 1000 1000 100 | = 0.99 | = 1.86 | = 3.79 | = 4.27 |
| 1000 1000 1000 | = 0.99 | = 1.96 | = 3.58 | = 4.31 |
| **Srednje ubrzanje** | **1.04** | **1.98** | **3.40** | **3.55** |

* + 1. Logovi izvršavanja

width=30, height=30, iteration=1000

Elapsed time is 34.440937 (sequencial) in ms

Elapsed time is 29.664579 (parallel) in ms

Test PASSED

width=500, height=500, iteration=10

Elapsed time is 71.316404 (sequencial) in ms

Elapsed time is 68.871844 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=100

Elapsed time is 2800.630897 (sequencial) in ms

Elapsed time is 2818.770994 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=1000

Elapsed time is 27516.952744 (sequencial) in ms

Elapsed time is 27719.500210 (parallel) in ms

Test PASSED

Listing 13. Sekvencijalna i paralelna izvršavanja sa 1 niti

width=30, height=30, iteration=1000

Elapsed time is 37.226502 (sequencial) in ms

Elapsed time is 18.768598 (parallel) in ms

Test PASSED

width=500, height=500, iteration=10

Elapsed time is 74.852194 (sequencial) in ms

Elapsed time is 35.637384 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=100

Elapsed time is 2792.358275 (sequencial) in ms

Elapsed time is 1499.991679 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=1000

Elapsed time is 27716.164482 (sequencial) in ms

Elapsed time is 14149.114480 (parallel) in ms

Test PASSED

Listing 14. Sekvencijalna i paralelna izvršavanja sa 2 niti

width=30, height=30, iteration=1000

Elapsed time is 34.902242 (sequencial) in ms

Elapsed time is 15.091054 (parallel) in ms

Test PASSED

width=500, height=500, iteration=10

Elapsed time is 71.264068 (sequencial) in ms

Elapsed time is 18.524861 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=100

Elapsed time is 2796.243935 (sequencial) in ms

Elapsed time is 737.566571 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=1000

Elapsed time is 27434.097856 (sequencial) in ms

Elapsed time is 7673.483783 (parallel) in ms

Test PASSED

Listing 15. Sekvencijalna i paralelna izvršavanja sa 4 niti

width=30, height=30, iteration=1000

Elapsed time is 36.417688 (sequencial) in ms

Elapsed time is 31.101013 (parallel) in ms

Test PASSED

width=500, height=500, iteration=10

Elapsed time is 73.424282 (sequencial) in ms

Elapsed time is 16.562246 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=100

Elapsed time is 2801.491332 (sequencial) in ms

Elapsed time is 656.605255 (parallel) in ms

Test PASSED

width=1000, height=1000, iteration=1000

Elapsed time is 27973.274361 (sequencial) in ms

Elapsed time is 6490.755107 (parallel) in ms

Test PASSED

Listing 16. Sekvencijalna i paralelna izvršavanja sa 8 niti

* + 1. Grafici ubrzanja

Slika 5. Grafik zavisnosti ubrzanja od broja niti

Slika 6. Grafik zavisnosti ubrzanja od broja niti sa worksharing direktivom i sa taskovima

* + 1. Diskusija dobijenih rezultata

Na osnovu dobijenih rezultata možemo videti da je postignuto ubrzanje sa paralelizacijom. Takođe sa povećanjem broja niti povećava se i ubrzanje. Isto možemo primetiti da u prvom test primeru, gde su dimenzije matrice manje, sa 8 niti imamo pad u ubrzanju u odnosu na manji broj niti, dok za ostale test primere imamo primetno veće ubrzanje Takođe nije bilo isplativo meriti vreme jednog izvršavanja funkcije evolve jer je ono previše malo, pa je mereno vreme samo na nivou cele simulacije.

Upoređivanjem rezultata trećeg i četvrtog problema možemo primetiti da za 1, 2 i 4 niti imamo identična ubrzanja sa worksharing direktivom i sa taskovima. Za 8 niti veće prosečno ubrzanje dobijamo worksharing direktivom, a ako bismo izuzeli prvi test primer onda su dobijena ubrzanja identična.

1. PROBLEM 5 – HOTSPOT
   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. Analizirati dati kod i obratiti pažnju na način izračunavanja temperatura. Ukoliko je potrebno međusobno isključenje prilikom paralelizacije programa, koristiti dostupne OpenMP konstrukte. Obratiti pažnju na efikasnost međusobnog isključenja niti i po potrebi ga svesti na što je moguće manju meru uvođenjem pomoćnih struktura podataka. Verifikaciju paralelizovanog rešenja vršiti nad dobijenim temperaturama u poslednjem stanju sistema. Način pokretanja programa se nalazi u datoteci run. [1,N]. Kao pomoćno sredstvo, data je i python skripta koja izlaznu datoteku formatira u heatmap sliku u PNG formatu.

* 1. Delovi koje treba paralelizovati
     1. Diskusija

U funkciji single\_iteration su uočene ugneždene dve for petlje cijom bi se paralelizacijom dobila optimizacija.

* + 1. Način paralelizacije

Imali smo dva moguća načina paralelizacije. Prvi način je bio pomoću taskova. Jedna od mogućnosti je bila da ceo unutrašnji for stavimo u task, dok je druga bila da telo unutrašnje petlje stavimo u task. Drugi način je bio pomoću worksharing direktive for i zajedno sa njim smo pokušali i sažimanje pomoću odredbe collapse. Najbolje vreme izvršavanja smo dobili pomoću worksharing direktive for bez collapse odredbe. Probali smo različite rasporede schedule direktive i zaključili da najbolja vremena dobijamo iz schedule(static).

* 1. Rezultati

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| broj niti  ulaz | 1 | 2 | 4 | 8 |
| 32 32 8192 1 ./data/temp32\_32 ./data/power32\_32 | = 1.12 | = 1.62 | = 2.20 | = 1.89 |
| 256 256 8192 1 ./data/temp256\_256 ./data/power256\_256 | = 1.04 | = 1.96 | = 3.30 | = 3.56 |
| 1024 1024 4096 1 ./data/temp1024\_1024 ./data/power1024\_1024 | = 1.01 | = 1.99 | = 3.70 | = 3.48 |
| 1024 1024 8192 1 ./data/temp1024\_1024 ./data/power1024\_1024 | = 1.00 | = 1.98 | = 3.70 | = 3.50 |
| 1024 1024 16384 1 ./data/temp1024\_1024 ./data/power1024\_1024 | = 1.01 | = 1.96 | = 3.72 | = 3.65 |
| 1024 1024 32768 1 ./data/temp1024\_1024 ./data/power1024\_1024 | = 1.00 | = 1.92 | = 3.81 | = 3.49 |
| **Srednje ubrzanje** | **1.03** | **1.91** | **3.40** | **3.26** |

* + 1. Logovi izvršavanja

grid\_rows=32, grid\_cols=32, sim\_time=8192, tempfile=./data/temp32\_32, powerfile=./data/power32\_32

Time elapsed in ms: 26.962247 (sequencial)

Time elapsed in ms: 24.150516 (parallel)

Test PASSED

grid\_rows=256, grid\_cols=256, sim\_time=8192, tempfile=./data/temp256\_256, powerfile=./data/power256\_256

Time elapsed in ms: 1310.529419 (sequencial)

Time elapsed in ms: 1255.654231 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=4096, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 10838.898952 (sequencial)

Time elapsed in ms: 10746.099056 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=8192, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 21431.406103 (sequencial)

Time elapsed in ms: 21381.387856 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=16384, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 42634.706423 (sequencial)

Time elapsed in ms: 42035.007131 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=32768, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 85310.948150 (sequencial)

Time elapsed in ms: 85257.660631 (parallel)

Test PASSED

Listing 17. Sekvencijalna i paralelna izvršavanja sa 1 niti

grid\_rows=32, grid\_cols=32, sim\_time=8192, tempfile=./data/temp32\_32, powerfile=./data/power32\_32

Time elapsed in ms: 25.911847 (sequencial)

Time elapsed in ms: 15.981831 (parallel)

Test PASSED

grid\_rows=256, grid\_cols=256, sim\_time=8192, tempfile=./data/temp256\_256, powerfile=./data/power256\_256

Time elapsed in ms: 1251.973345 (sequencial)

Time elapsed in ms: 637.061000 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=4096, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 10992.281080 (sequencial)

Time elapsed in ms: 5525.470967 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=8192, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 21354.894798 (sequencial)

Time elapsed in ms: 10743.170405 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=16384, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 42821.434601 (sequencial)

Time elapsed in ms: 21777.588395 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=32768, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 85303.176833 (sequencial)

Time elapsed in ms: 44369.678572 (parallel)

Test PASSED

Listing 18. Sekvencijalna i paralelna izvršavanja sa 2 niti

grid\_rows=32, grid\_cols=32, sim\_time=8192, tempfile=./data/temp32\_32, powerfile=./data/power32\_32

Time elapsed in ms: 26.709957 (sequencial)

Time elapsed in ms: 12.117223 (parallel)

Test PASSED

grid\_rows=256, grid\_cols=256, sim\_time=8192, tempfile=./data/temp256\_256, powerfile=./data/power256\_256

Time elapsed in ms: 1259.812312 (sequencial)

Time elapsed in ms: 381.035069 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=4096, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 10621.652955 (sequencial)

Time elapsed in ms: 2868.603117 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=8192, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 21111.068628 (sequencial)

Time elapsed in ms: 5705.231245 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=16384, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 42338.025070 (sequencial)

Time elapsed in ms: 11381.782590 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=32768, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 84609.990933 (sequencial)

Time elapsed in ms: 22165.098460 (parallel)

Test PASSED

Listing 19. Sekvencijalna i paralelna izvršavanja sa 4 niti

grid\_rows=32, grid\_cols=32, sim\_time=8192, tempfile=./data/temp32\_32, powerfile=./data/power32\_32

Time elapsed in ms: 26.236656 (sequencial)

Time elapsed in ms: 13.823957 (parallel)

Test PASSED

grid\_rows=256, grid\_cols=256, sim\_time=8192, tempfile=./data/temp256\_256, powerfile=./data/power256\_256

Time elapsed in ms: 1307.613160 (sequencial)

Time elapsed in ms: 367.146718 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=4096, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 10610.959799 (sequencial)

Time elapsed in ms: 3045.354849 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=8192, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 21352.541808 (sequencial)

Time elapsed in ms: 6107.724201 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=16384, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 44829.628456 (sequencial)

Time elapsed in ms: 12285.997582 (parallel)

Test PASSED

grid\_rows=1024, grid\_cols=1024, sim\_time=32768, tempfile=./data/temp1024\_1024, powerfile=./data/power1024\_1024

Time elapsed in ms: 85881.186900 (sequencial)

Time elapsed in ms: 24619.479462 (parallel)

Test PASSED

Listing 20. Sekvencijalna i paralelna izvršavanja sa 8 niti

* + 1. Grafici ubrzanja

Slika 7. Grafik zavisnosti ubrzanja od broja niti

* + 1. Diskusija dobijenih rezultata

Uočava se ubrzanje za paralelni program u odnosu na sekvencijalni, dok za 2, 4 i 8 niti to ubrzanje je znatno veće. Primećujemo da je najbolje ubrzanje postignuto sa 4 niti, a nakon toga sa 8 niti.