Univerzitet u Beogradu - Elektrotehnički fakultet

Multiprocesorki sistemi (13S114MUPS, 13E114MUPS)



**Domaći zadatak 2 – MPI**

Izveštaj o urađenom domaćem zadatku

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

Beograd, novembar 2020.

**Sadržaj**

[***Problem 1 – Pi Calc bez podele posla***](#_heading=h.goxsgvm9mb7y) ***3***

[*Tekst problema*](#_heading=h.1fob9te) *3*

[*Diskusija*](#_heading=h.2et92p0) *3*

[*Način paralelizacije*](#_heading=h.tyjcwt) *3*

[*Rezultati*](#_heading=h.3dy6vkm) *3*

[*Logovi izvršavanja*](#_heading=h.1t3h5sf) *3*

[*Grafici ubrzanja*](#_heading=h.4d34og8) *6*

[*Diskusija dobijenih rezultata*](#_heading=h.wdhcxoaaeytx) *7*

[***Problem 2 – Needleman-Wunsch***](#_heading=h.6w7qxg4edr5c) ***8***

[*Tekst problema*](#_heading=h.v529r8twir76) *8*

[*Diskusija*](#_heading=h.nofe6n77aa4g) *8*

[*Način paralelizacije*](#_heading=h.t6kjmf7qslk3) *8*

[*Rezultati*](#_heading=h.apmny2pou401) *8*

[*Logovi izvršavanja*](#_heading=h.5fl4u9rihv44) *8*

[*Grafici ubrzanja*](#_heading=h.qep3d97znuuk) *11*

[*Diskusija dobijenih rezultata*](#_heading=h.dxzqpdubze22) *11*

[***Problem 5 – Nbody***](#_heading=h.tgyrkrppxok2) ***11***

[*Tekst problema*](#_heading=h.gsabqr8t8jha) *11*

[*Diskusija*](#_heading=h.4w3pfa39q2sp) *12*

[*Način paralelizacije*](#_heading=h.xpzc0bnk47i) *12*

[*Rezultati*](#_heading=h.okvdgx34yh3o) *12*

[*Logovi izvršavanja*](#_heading=h.bdfg7hzc183w) *12*

[*Grafici ubrzanja*](#_heading=h.4v746fsn2fxa) *13*

[*Diskusija dobijenih rezultata*](#_heading=h.1xplgymh7u7w) *14*

# Problem 1 – Pi Calc bez podele posla

## Tekst problema

Paralelizovati program koji izračunava vrednost broja PI korišćenjem formule: 𝜋 = 4 ∗ ∑ (−1) 𝑘+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. 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]

### Diskusija

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

### Način paralelizacije

Na početku paralelnog programa je prosleđena vrednost n, odnosno zadat broj tacaka, svim procesima metodom *Broadcast* a zatim su poslovi raspodeljeni ravnopravno tako da svaki proces obrađuje tačke sa razmakom od *size*koji predstavlja broj procesora u grupi. Na kraju se izračunate vrednosti sakupe metodom *Reduce*.

## Rezultati

### Logovi izvršavanja

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

Za n = 2:

**SEQUENTIAL:**

**Before for loop, factor = 0.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165358977**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.004628**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165358971**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.001734**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255358979**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.032388**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255358581**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.015930**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264358933**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.318343**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264358449**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.159266**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265258805**

**Ref estimate of pi = 3.14159265358979**

**The process took: 3.194186**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265258555**

**Ref estimate of pi = 3.14159265358979**

**The process took: 1.592206**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265348835**

**Ref estimate of pi = 3.14159265358979**

**The process took: 32.044826**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265348480**

**Ref estimate of pi = 3.14159265358979**

**The process took: 15.921904**

**TEST PASSED**

**Listing 1. Sekvencijalna i paralelna izvršavanja Pi Calc za n=2**

Za n = 4:

**SEQUENTIAL:**

**Before for loop, factor = 0.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165358977**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.003326**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165359035**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.001638**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255358979**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.033102**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255359074**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.016037**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264358933**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.321542**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264359661**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.079622**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265258805**

**Ref estimate of pi = 3.14159265358979**

**The process took: 3.238965**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265259144**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.796466**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265348835**

**Ref estimate of pi = 3.14159265358979**

**The process took: 31.842533**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265349118**

**Ref estimate of pi = 3.14159265358979**

**The process took: 7.968353**

**TEST PASSED**

**Listing 1. Sekvencijalna i paralelna izvršavanja Pi Calc za n= 4**

Za n = 8:

**SEQUENTIAL:**

**Before for loop, factor = 0.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165358977**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.003240**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000 terms**

**Our estimate of pi = 3.14159165358956**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.001295**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255358979**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.032286**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000 terms**

**Our estimate of pi = 3.14159255358947**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.008129**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264358933**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.323165**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 100000000 terms**

**Our estimate of pi = 3.14159264359058**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.079905**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265258805**

**Ref estimate of pi = 3.14159265358979**

**The process took: 3.230972**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 1000000000 terms**

**Our estimate of pi = 3.14159265259308**

**Ref estimate of pi = 3.14159265358979**

**The process took: 0.800090**

**TEST PASSED**

**SEQUENTIAL:**

**Before for loop, factor = 1.000000.**

**After for loop, factor = -1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265348835**

**Ref estimate of pi = 3.14159265358979**

**The process took: 34.795706**

**PARALLEL:**

**Before for loop, factor = -1.000000.**

**After for loop, factor = 1.000000.**

**With n = 10000000000 terms**

**Our estimate of pi = 3.14159265349295**

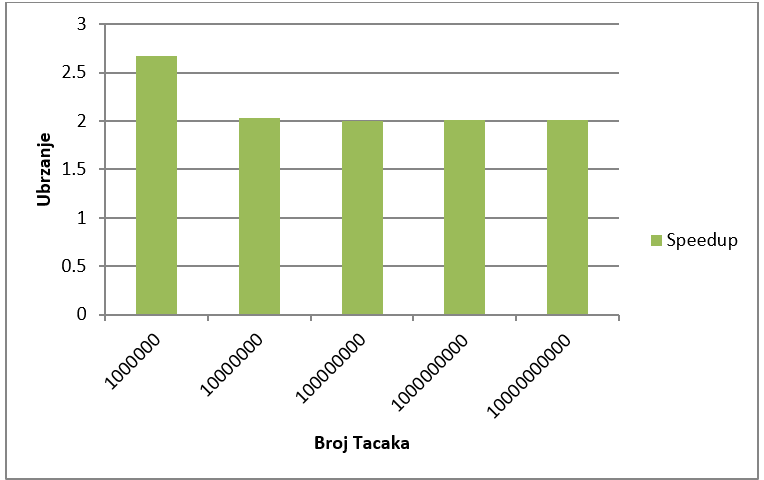
**Ref estimate of pi = 3.14159265358979**

**The process took: 8.036458**

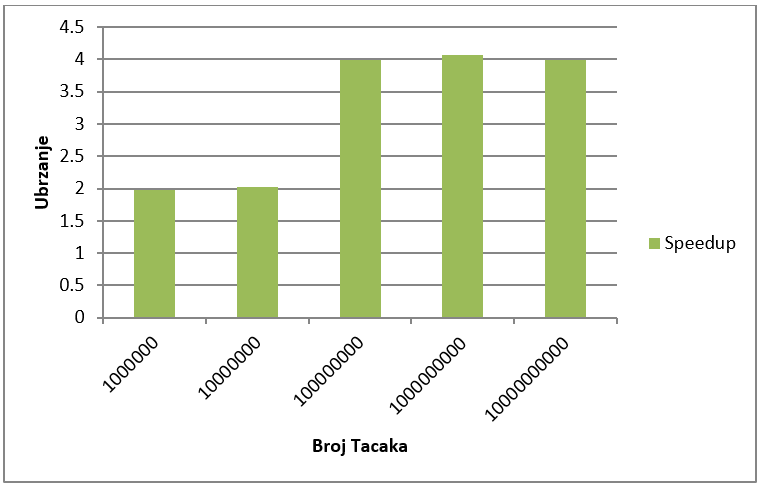
**TEST PASSED**

**Listing 1. Sekvencijalna i paralelna izvršavanja Pi Calc za n = 8**

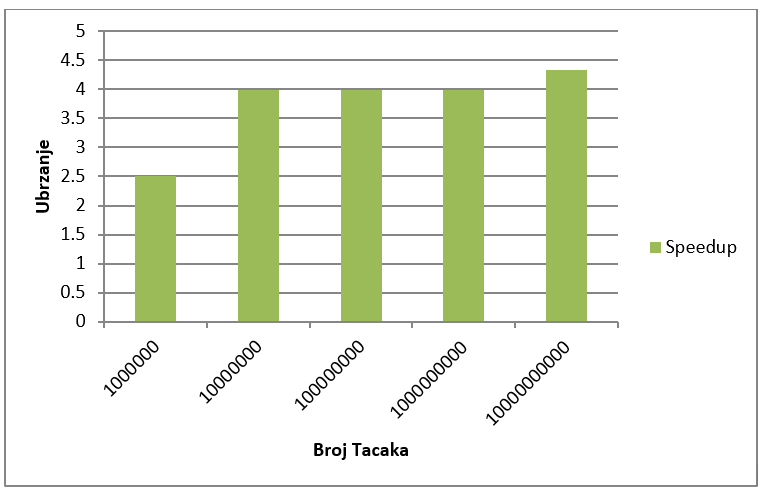
### Grafici ubrzanja



**Grafik zavisnosti ubrzanja za N = 2 niti**

****

**Grafik zavisnosti ubrzanja za N = 4 niti**

****

**Grafik zavisnosti ubrzanja za N = 8 niti**

### Diskusija dobijenih rezultata

Za 2 procesora je se pokazuje da je ubrzanje veće za manji broj tačaka dok za veći broj efikasnost opada. Sa druge strane, sa povećanjem broja procesora povećava i ubrzanje za veci broj tačaka. Sa 8 niti se dostiže najveće ubrzanje za 100000000000 tačaka od oko 4.3 puta u odnosu na sekvencijalno izvršavanje.

# Problem 2 – Needleman-Wunsch

## 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. Paralelizaciju pokušati podelom sekvence za poravnavanje na podsekvence. Ukoliko je moguće, koristiti rutine za neblokirajuću komunikaciju za razmenu poruka. Program testirati sa parametrima koji su dati u datoteci run. [1, N]

### Diskusija

Ovaj problem ima 2 krucijalne petlje koje se mogu paralelizovati i obavljaju glavni deo računanja za ovaj algoritam.

### Način paralelizacije

Pralelizacija se postiže raspodelom posla prilikom određivanja vrednosti na dijagonali. Prvo se izračunaju vrednosti u jednoj diagonali potom se te vrednosti koriste za računanje sledeće dijagonale. Međutim, kako MPI nije zasnovan na deljenoj memoriji mora da se obezbedi svakom čvoru da poseduje sve potrebne ulazne vrednosti za računanje sledećeg polja u matrici. Ovo se radi direktnom komunikacijom izmedju čvorova tako što nakon određivanja vrednosti jedan čvor šalje poruku čvoru koji ima rag za jedan veći (ili 0 za čvor sa največim rangom), a pre računanja prima vrednost od čvora koji ima rang za jedan manji (odnosno najveći za čvor sa rangom 0). Iako je za računanje potrebno posedovati tri ulazne promenjive samo jedna nije posedovana a to je vrednost polja matrice koje je tačno iznad polja koje se trenutno računa, promenjiva koja se nalazi levo od polja koje se računa je već ranije izračunata od stran baš tog čvora (ili generisana) a polje dijagonalno tj. iznad i levo je već bilo prosleđeno u prethodnom koraku (ili generisano).

## Rezultati

### Logovi izvršavanja

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

**2048 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 0.283193**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 1.937992**

**6144 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 2.327414**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 21.324301**

**16384 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 18.881097**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 205.516208**

**22528 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 40.178664**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 440.340623**

**Listing 1. Sekvencijalno izvršavanje Needleman-Wunsch n=2**

**2048 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 0.193477**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 1.100761**

**6144 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 1.604836**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 11.983266**

**16384 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 17.188613**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 113.923967**

**22528 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 44.724238**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 248.750683**

**Listing 1. Sekvencijalno izvršavanje Needleman-Wunsch n=3**

**2048 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 0.162381**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 0.844474**

**6144 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 1.444147**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 9.026282**

**16384 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 11.756897**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 89.445972**

**22528 10**

**SEQUENTIAL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

**Processing bottom-right matrix**

**The process took: 73.551236**

**PARALLEL**

**Start Needleman-Wunsch**

**Processing top-left matrix**

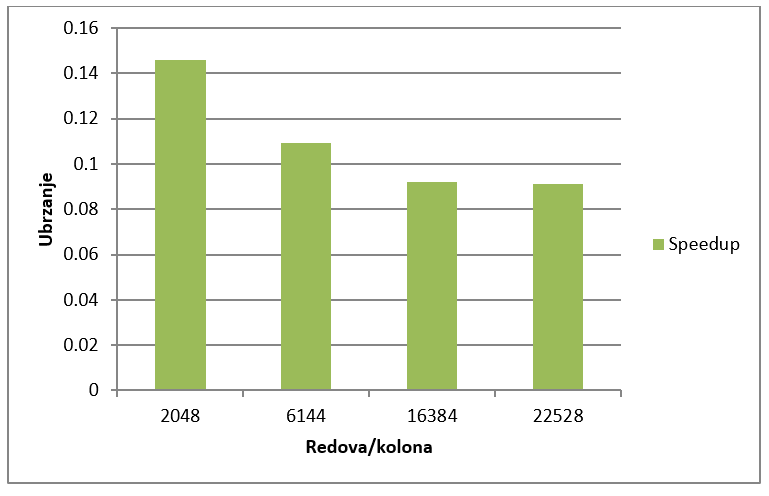
**Processing bottom-right matrix**

**The process took: 368.377856**

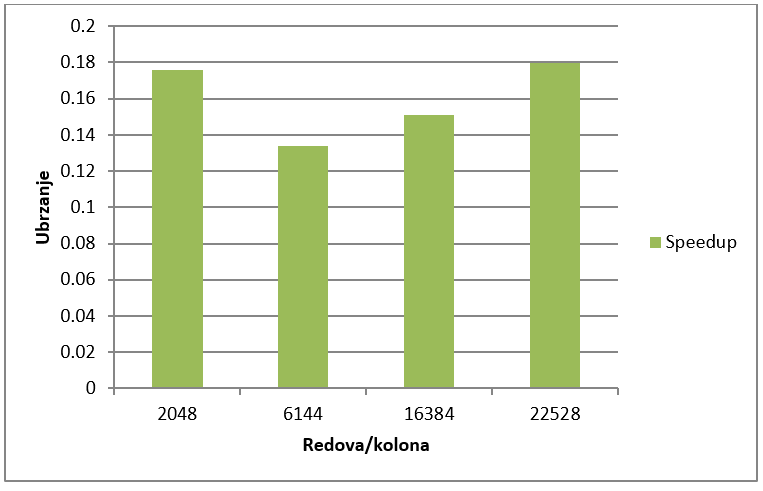
**Listing 1. Sekvencijalno izvršavanje Needleman-Wunsch n=4**

### Grafici ubrzanja

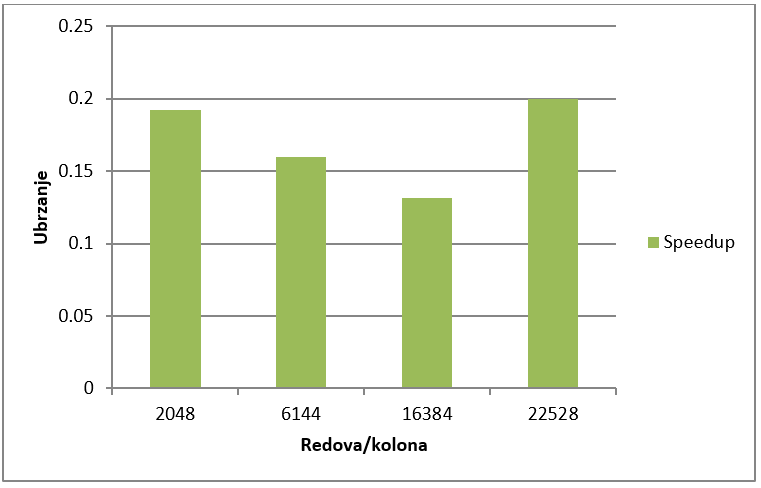
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



**Grafik zavisnosti ubrzanja za N = 2 niti**

****

**Grafik zavisnosti ubrzanja za N = 3 niti**

****

**Grafik zavisnosti ubrzanja za N = 4 niti**

### Diskusija dobijenih rezultata

Za sve test primere ubrzanje paralelnog programa iznosi izmedju 0.08 i 0.16 u odnosu na sekvencijalni program. Predlozeno resenje previse vremena odvaja za komunikaciju pa režijski troškovi brzo prevazilaze dobizak od paralelizacije.

# Problem 3 – Nbody

## 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. Verifikaciju paralelizovanog rešenja vršiti nad dobijenim energijama i poslednjem stanju sistema. Način pokretanja programa se nalazi u datoteci run. [1, N]

### 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.

### Način paralelizacije

Na početku se vrši generisanje čestica samo sa jednom *master* niti, nakon čega se niz čestica pošalje svim ostalim nitima pomoću *Broadcast* funkcije.

Petlja za računanje kinetičke energije se optimizuje korišćenjem aditivne redukcije nad ukupnom kinetičkom energijom. Petlja za računanje potencijalne energije se optimizuje ali samo spoljašnja na taj način da se poslovi raspodele ravnopravno tako da svaki procesor obrađuje tačke sa razmakom od *size*koji predstavlja broj procesora u grupi.

Takodje se optimizuje poziv funkcije Compute\_force funkcij i na isti način raspodeli posao kao za potencijalnu energiju. Nakon izračunavanja svih sila za zadate čestice u odgovarajućim čvorovima vrši se redukcija sumiranjem svih izračunatih vrednosti sila svih čestica kako bi se dobile konačne vrednosti za taj korak.

## Rezultati

### Logovi izvršavanja

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

N = 2

**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: 0.036318**

**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: 0.045537**

**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: 0.798241**

**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: 0.546872**

**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: 82.747269**

**PARELLEL**

**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: 46.696617**

**Test PASSED**

N = 4

**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: 0.033947**

**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: 0.065080**

**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: 0.802244**

**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: 0.492022**

**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: 80.116045**

**PARELLEL**

**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: 23.230467**

**Test PASSED**

N = 8

**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: 0.036321**

**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: 0.243822**

**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: 0.905706**

**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: 1.241671**

**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: 90.680277**

**PARELLEL**

**PE = -6.751832e+38, KE = 1.125000e+37, Total Energy = -6.639332e+38**

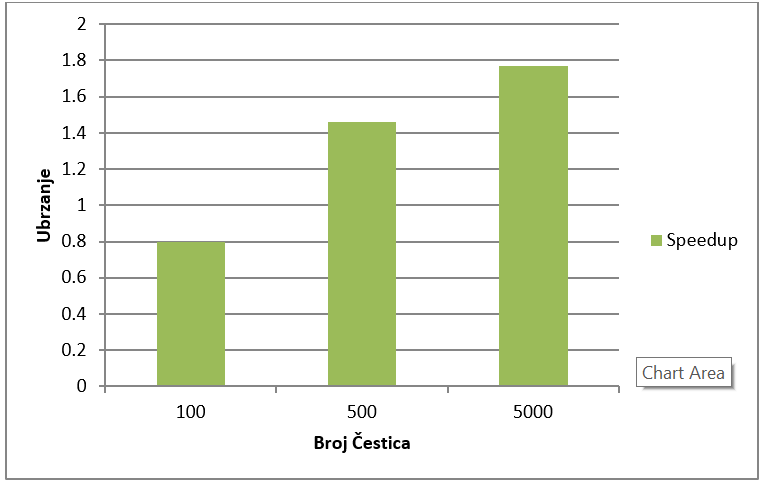
**PE = -6.649074e+38, KE = 2.481116e+36, Total Energy = -6.624263e+38**

**The process took: 31.646480**

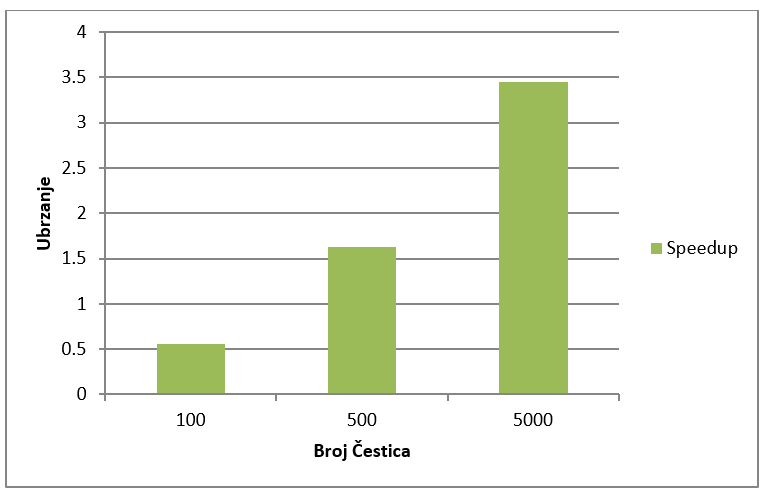
**Test PASSED**

### Grafici ubrzanja

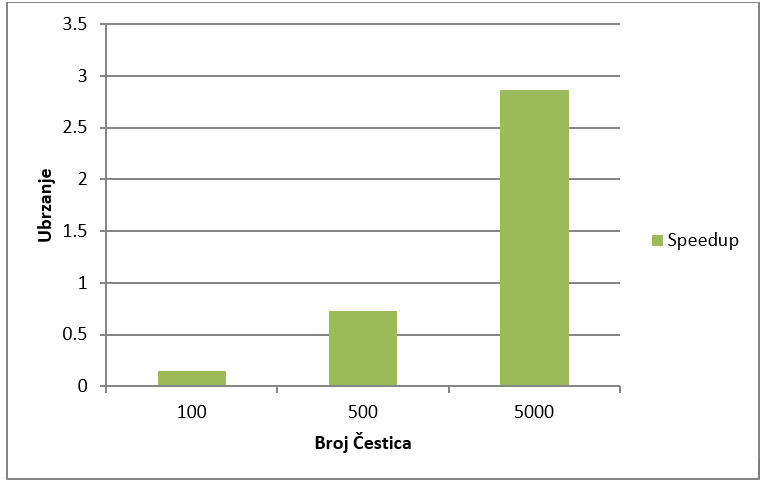
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



**Grafik zavisnosti ubrzanja za N = 2 niti (primer)**



**Grafik zavisnosti ubrzanja za N = 4 niti (primer)**



**Grafik zavisnosti ubrzanja za N = 8 niti (primer)**

### Diskusija dobijenih rezultata

Za sva tri primera nezavisno od broja procesora evidentno je da se ubrzanje pocećava sa povećanjem broja čestica koje se obrađuju. Takođe se može primetiti da se za mali broj čestica ne dobija ubrzanje već usporenje.

Za 2 procesora se ni sa povećanjem broja čestica ne dobija veliko ubrzanje svega oko 50% dok se sa 4 i 8 procesora može postići oko 3 puta veće ubrzanje za 5000 čestica.

# Problem 4 – Nbody sa manager-worker

## Tekst problema

Prethodni program paralelizovati korišćenjem manager - worker modela u delu proračuna sila. Proces gospodar (master) treba da učita neophodne podatke, generiše poslove, deli posao ostalim procesima i ispiše na kraju dobijeni rezultat. U svakom koraku obrade, proces gospodar šalje procesu radniku na obradu jednu jedinicu posla čiji veličinu treba pažljivo odabrati. Proces radnik prima podatke, vrši obradu, vraća rezultat, signalizira gospodaru kada je spreman da primi sledeći posao i ponavlja opisani postupak dok ne dobije signal da prekine sa radom. Veličinu jedne jedinice posla prilagoditi karakteristikama programa. Ukoliko je moguće, koristiti rutine za neblokirajuću komunikaciju za razmenu poruka. 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. Verifikaciju paralelizovanog rešenja vršiti nad dobijenim energijama i poslednjem stanju sistema. Način pokretanja programa se nalazi u datoteci run. [1, N]

### Diskusija

Kao što se i zahteva u zadatku paralelizacija pomoću master-slave modela se jedino vrši za računanje sila. Sve ostale optimizacije su iste kao i u rešenju zadatka 3.

### Način paralelizacije

Ovde master nit deli poslove slave nitima sa nekom zadatom veličinom paketa posla (grain\_size). U ovaj režim se ulazi u svakoj iteraciji petlje koja prolazi kroz svaki korak simulacije. Komunikacija se vrši slanjem strukture koja definiše posao pomoću MPISend i MPIRecv funkcije. Prilikom definisanja posla ne prenosi se niz svih čeestica sistema jer to svi čvorovi poseduju već samo specificira koji deo posla slave treba da obavi. Rezultati se upisuju u lokalni niz sila koji se zatim redukuje.

## Rezultati

### Logovi izvršavanja

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

N = 2

**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: 0.036168**

**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: 0.036713**

**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: 0.799098**

**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: 0.662632**

**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: 79.987579**

**PARELLEL**

**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: 60.800580**

**Test PASSED**

N = 4

**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: 0.034046**

**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: 0.036909**

**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: 0.799213**

**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: 0.370956**

**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: 80.086250**

**PARELLEL**

**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: 24.709014**

**Test PASSED**

N = 8

**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: 0.036278**

**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: 0.075854**

**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: 0.904720**

**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: 0.594009**

**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: 91.192575**

**PARELLEL**

**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: 24.674364**

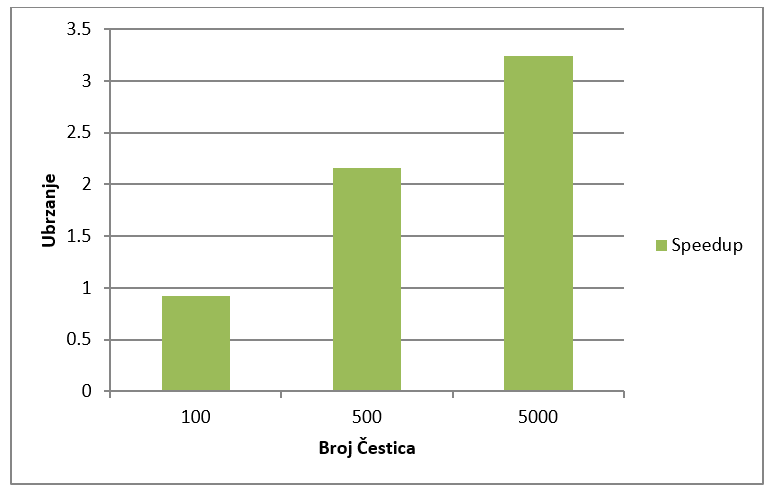
**Test PASSED**

### Grafici ubrzanja

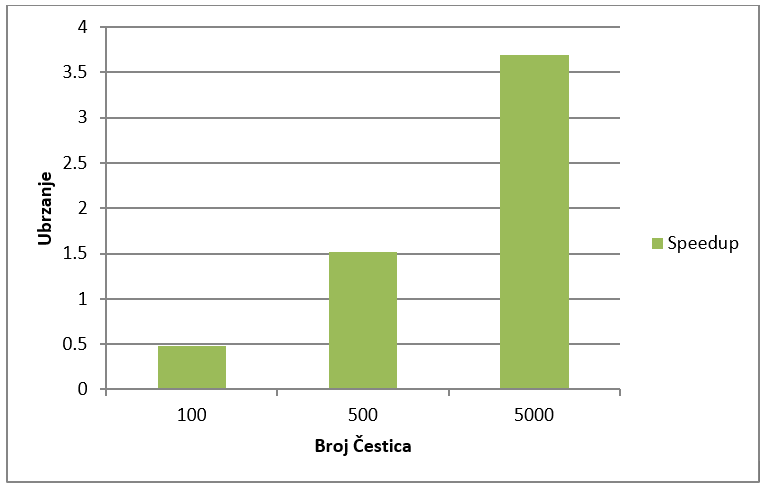
U okviru ove sekcije su dati grafici ubrzanja u odnosu na sekvencijalnu implementaciju.



**Grafik zavisnosti ubrzanja za N = 2 niti (primer)**

****

**Grafik zavisnosti ubrzanja za N = 4 niti (primer)**

****

**Grafik zavisnosti ubrzanja za N = 8 niti (primer)**

### Diskusija dobijenih rezultata

U odnosu na prethodni zadatak, dobijaju se bolji rezultati za mali broj čestica. Za 500 čestica se ovde dobija ubrzanje i sa 8 procesora što nije bio slučaj u prethodnom zadatku i maksimum ubranja se dobija za 5000 čestica sa 8 procesora i to više od 3.5 puta što je više i od najboljeg ubrzanja prethodnog primera. Isprobane su različite vrednosti grainsize parametra i nisu dobijene znatne promene. Najveće ubrzanje smo dobili za grain\_size 10 međutim razlike u brzini nisu dovoljno velike da bi se empirijski napravio zaključak.