Vilniaus universitetas
Matematikos ir informatikos fakultetas
Programų sistemų katedra

Lygiagrečiojo programavimo

Laboratorinio darbo #1 ataskaita

Autorius: 3 kursas, 5 grupė
Mantas Petrikas

Vilnius, 2017

Užduotis

1. Parsisiųskite ir sukurtame kataloge išsaugokite du failus: algoritmo kodą ir duomenų failą.
2. Įvertinkite algoritmo teorinį pagreitėjimą naudojant $p \in \{1, 2, 4\}$ gijų, kai uždavinio dydis $N \in \{24000, 48000, 96000\}$.
3. Išlygiagretinkite funkcija *performcalc* taip, kad kiekviena gija atliktu skaičiavimus su jai priskirtu duomenų bloku. Ciklo FOR lygiagretinimui naudokite

```
for (int i=id*chunk; i<(id+1)*chunk; i++)
```

Čia id – gijos ID, chunk – darbo dalis (iteracijos) skirta vienai gijai.

4. Atlikite lygiagrečiuosius skaičiavimus naudodami $p \in \{1, 2, 4\}$ gijas fiksuodami nuosekliosios dalies, lygiagrečiosios dalies ir bendra algoritmo pagreitėjimą.
5. Palyginkite eksperimentiniu būdu gautus rezultatus su teoriniais įverčiais.

Gauti rezultatai

Norint įvertinti algoritmo teorinį pagreitėjimą neišlygiagretintas algoritmas buvo po tris kartus leidžiamas prisijungus prie MIF linux serverių su skirtingais uždavinio dydžiams N. Gauti vidutiniai rezultatai pateikiami 1 lentelėje. Bandymų metu buvo prisijungta prie kompiuterio turinčio 4 branduolius.

1 lentelė. Nelygiagretinto algoritmo vykdymo vidutiniai rezultatai.

N dydis	Nuosekliosios algoritmo dalies vykdymo laikas (s)	Lygiagretinamosios algoritmo dalies vykdymo laikas (s)	Nuosekliosios algoritmo dalies vykdymo laiko dalis α	Lygiagretinamosios algoritmo dalies vykdymo laiko dalis β
24000	0,97	10,74	0,0828	0,9172
48000	1,92	42,95	0,0428	0,9572
96000	3,82	172,08	0,0217	0,9783

Lygiagrečiojo algoritmo teorinis pagreitėjimas S_p naudojant p procesorių apskaičiuotas naudojant formulę

$$\tilde{S}_p = \frac{1}{\alpha + \frac{\beta}{p}}.$$

Teoriniai lygiagrečiojo algoritmo vykdymo laikas T_p naudojant p procesorių skaičiuotas naudojant formulę

$$T_p = \frac{T_0}{S_p},$$

kur T_0 - nuosekliojo algoritmo vykdymo trukmė.

Teoriniai lygiagrečiojo algoritmo pagreitėjimai ir vykdymo trukmės naudojant 1, 2 ar 4 procesorius pateikiamos 2 lentelėje.

2 lentelė. Teoriniai lygiagrečiojo algoritmo pagreitėjimo įverčiai ir vykdymo trukmės.

N dydis	S_1	T_1 (s)	S_2	T_2 (s)	S_4	T_4 (s)
24000	1	11,71	1,85	6,34	3,20	3,65
48000	1	44,87	1,92	23,40	3,54	12,66
96000	1	175,90	1,96	89,86	3,76	46,84

S_p - teorinis lygiagrečiojo algoritmo pagreitėjimo įvertis naudojant p procesorių.
T_p - teorinė lygiagrečiojo algoritmo vykdymo trukmė naudojant p procesorių.

Išlygiagretinus procedūrą *performcalc*, atlikti programos kodas po tris kartus vykdytas naudojant 1, 2 ir 4 procesorius. Nuosekliosios ir lygiagrečiosios algoritmo dalies vidutinės vykdymo trukmės naudojant skirtingus duomenų kiekius N pateikiami 3 lentelėje.

3 lentelė. Nuosekliosios ir lygiagrečiosios algoritmo dalies vykdymo trukmės.

N dydis	1 procesorius		2 procesoriai		4 procesoriai	
	Nuos.	Lyg.	Nuos.	Lyg.	Nuos.	Lyg.
24000	0,99	12,14	0,99	6,09	0,99	3,05
48000	1,99	48,64	1,98	24,18	1,99	12,23
96000	3,99	191,42	4,00	97,03	3,99	49,88

Nuos. – Nuosekliosios algoritmo dalies vykdymo trukmė (s).
Lyg. – Lygiagrečiosios algoritmo dalies vykdymo trukmė (s).

Praktiniai pagreitėjimai S, apskaičiuoti pagal formulę

$$S_p = \frac{T_0}{T_p},$$

kur T_0 – nuosekliojo algoritmo vykdymo laikas, T_p – lygiagretaus algoritmo vykdymo laikas naudojant p procesorių, pateikiami 4 lentelėje.

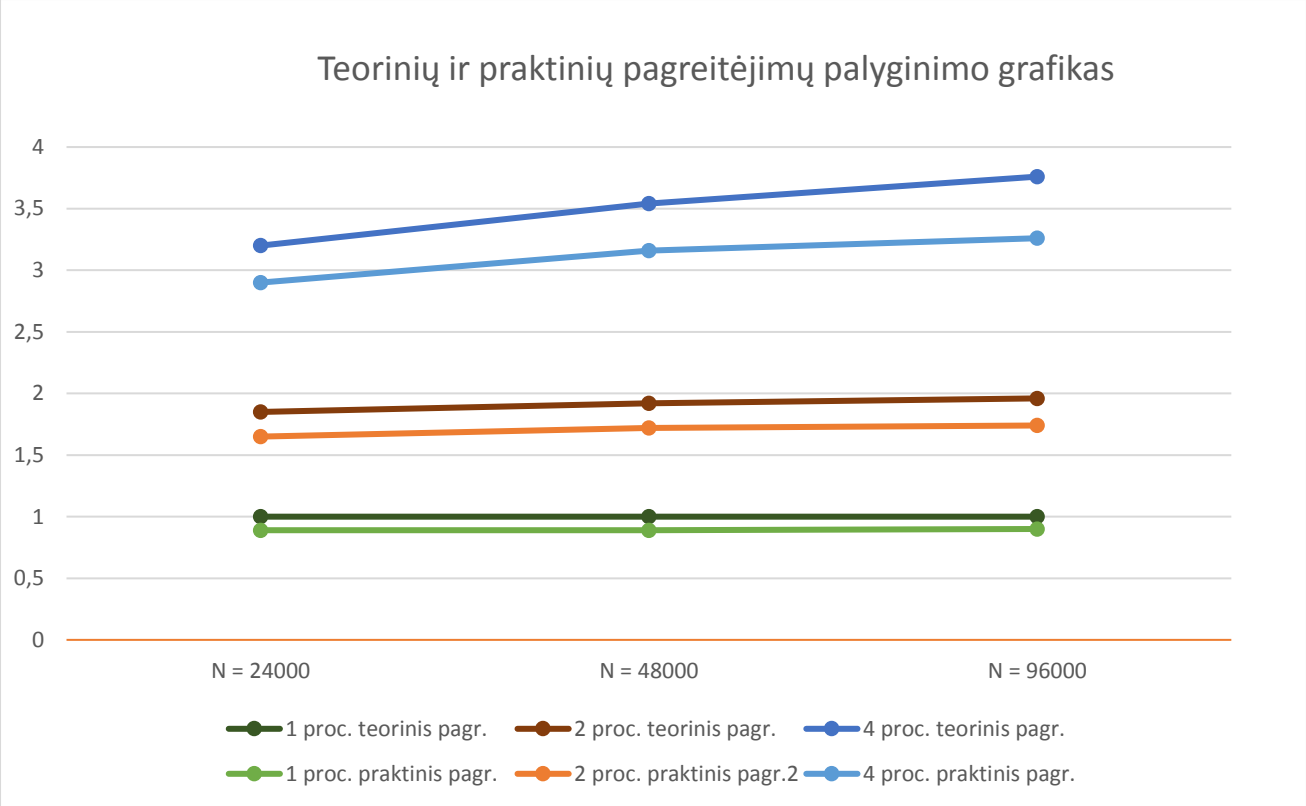
4 Lentelė. Praktiniai algoritmo pagreitėjimų įverčiai.

N dydis	Naudotų procesorių kiekis	Nuosekliosios dalies pagreitėjimas	Lygiagrečiosios dalies pagreitėjimas	Bendras algoritmo pagreitėjimas
24000	1	0,98	0,88	0,89
48000	1	0,96	0,88	0,89
96000	1	0,96	0,90	0,90
24000	2	0,98	1,76	1,65
48000	2	0,97	1,78	1,72
96000	2	0,96	1,77	1,74
24000	4	0,98	3,52	2,90
48000	4	0,96	3,51	3,16
96000	4	0,96	3,45	3,26

Palyginus teorinius ir praktinius pagreitėjimo įverčius matoma, kad praktinis pagreitėjimas 10 – 13 procentų mažesnis už teorinį. Pagreitėjimo skirtumai pateikiami 5 lentelėje ir 1 grafike.

5 lentelė. Teorinių ir praktinių pagreitėjimų palyginimas

N dydis	Naudotų procesorių Kiekis	Teorinis pagreitėjimas	Praktinis pagreitėjimas	Teorinio ir praktinio pagreitėjimo įverčių skirtumas.	Nuokrypis nuo teorinių pagreitėjimo (%)
24000	1	1	0,89	0,11	11,00%
48000	1	1	0,89	0,11	11,00%
96000	1	1	0,90	0,10	10,00%
24000	2	1,85	1,65	0,20	10,81%
48000	2	1,92	1,72	0,20	10,42%
96000	2	1,96	1,74	0,22	11,22%
24000	4	3,20	2,90	0,30	9,38%
48000	4	3,54	3,16	0,38	10,73%
96000	4	3,76	3,26	0,50	13,30%



1 grafikas. Teoriniai ir praktiniai algoritmo pagreitėjimai naudojant skirtingą procesorių kiekį.

Išvados

Palyginus teorinius algoritmo pagreitėjimo įverčius su realiais, matyti, kad praktiniai įverčiai yra 10-16% mažesni už teorinius. Neatitikimai galėjo susidaryti nes teoriniai įverčiai neatsižvelgia į papildomą darbą atliekamą lygiagretinant kodą ir trikdžių eksperimento metu. Taip pat matyti kad kuo didesnė uždavimo dalį sudaro lygiagretinama dalis tuo didesnis bendras algoritmo pagreitėjimas, bet tuo didesnis nuokrypis nuo teorinio pagreitėjimo.
Šio eksperimento metu pasidarė akivaizdu kad lygiagretus algoritmas vykdomas naudojant 1 procesorių gali būti mažesnis už tomis pačiomis sąlygomis vykdomą tą patį nuoseklų algoritmą.

Priedai

Išlygiagretinta procedūra *performcalc:*

```
void performcalc(int N, int p, float* M, float* D) {

   #pragma omp parallel
   {
      int threadCount = omp_get_num_threads();
      int threadId = omp_get_thread_num();
      int chunk = N/threadCount;
      float min, d;
      for (int i=threadId*chunk; i<(threadId+1)*chunk; i++) {
         min = 1e10;
         for (int j=0; j<N; j++) {
            if (j != i) {
               d = (M[2*i]-M[2*j]) + (M[2*i+1]-M[2*j+1]);
               if (d < min) {
                  min = d;
               }
            }
            D[i] = min;
         }
      }
   }
}
```

Visas programos kodas ir eksperimentų rezultatai pasiekiami internetiniu adresu
http://www.github.com/BinaryHydra/parallel-programing