

PRIR – Ćwiczenie 4

Mateusz Ciupa (291062)
Krzysztof Mózdyński (290847)

Wstęp

Sekwencyjny kod programu został zrównoleglony przy pomocy OpenMP. Części programu, które uległy zrównolegleniu to druga faza wyszukiwania liczb pierwszych oraz ich wypisanie do pliku tekstowego. Sekcja krytyczna została użyta przy wpisywaniu liczb pierwszych do tablicy. Do zmierzenia czasu zastosowana została funkcja `clock()` z biblioteki standardowej języka C.

Pomiary

Wyniki pomiarów czasu dla programu sekwencyjnego (w sekundach):

n	time
10^5	0.037040
10^6	0.218287
10^7	3.905952

Wyniki pomiarów czasu dla programu zrównoleglonego (w sekundach):

Liczba wątków/n	10^5	10^6	10^7
2	0.014576	0.161228	3.693703
3	0.015194	0.255026	5.19646
4	0.030496	0.366161	7.929487
5	0.026694	0.306748	7.327469
6	0.05875	0.348035	7.452386
7	0.025195	0.351095	7.458823
8	0.030964	0.352346	7.324127

Przyspieszenie programu zostało obliczone za pomocą wzoru:

$$S_p = \frac{T_1}{T_p}$$

gdzie:

p – liczba procesorów

T_1 – czas wykonania dla algorytmu sekwencyjnego

T_p – czas wykonania dla algorytmu równoległego przy użyciu p procesorów.

Przyspieszenie zrównoleglonego programu:

Liczba wątków/n	10^5	10^6	10^7
2	2.54116355653128	1.35390254794453	1.05746238936915
3	2.43780439647229	0.855940178648451	0.751656319879302
4	1.21458551941238	0.596150327315034	0.492585712039127
5	1.38757773282386	0.711616701657387	0.533056093447819
6	0.630468085106383	0.62719841395262	0.524120999636895
7	1.47013296288946	0.621732009854883	0.523668680702036
8	1.1962278775352	0.619524558246723	0.533299327005116

Efektywność programu została obliczona za pomocą wzoru:

$$E_p = \frac{S_p}{p}$$

gdzie:

S_p – przyspieszenie programu

p – liczba procesorów

Efektywność zrównoleglonego programu:

Liczba wątków/n	10^5	10^6	10^7
2	1.27058177826564	0.676951273972263	0.528731194684575
3	0.812601465490764	0.285313392882817	0.250552106626434
4	0.303646379853095	0.149037581828758	0.123146428009782
5	0.277515546564771	0.142323340331477	0.106611218689564
6	0.105078014184397	0.104533068992103	0.087353499939483
7	0.210018994698495	0.088818858550698	0.074809811528862
8	0.1495284846919	0.077440569780841	0.06666241587564

Wnioski

Uznaliśmy, że nieefektywnym jest zrównoleglenie pierwszej fazy programu, ponieważ różnice czasowe między sekwencyjnym i równoległym wykonaniem tej części są rzędu 0,001s.

Czas wraz ze wzrostem rozmiaru zadania rośnie dla programu sekwencyjnego, jak i równoległego.

Czas w zależności od liczby wykorzystanych wątków rośnie dla liczby wątków od 2 do 4, natomiast w przypadku liczby wątków 5-8 oscyluje w określonym obszarze. Świadczy to o tym, że dla małej liczby wątków zrównoleglenie programu nie jest efektywne, natomiast dla większej już może być.

Przyspieszenie oraz efektywność programu niezależnie od liczby elementów wraz ze wzrostem liczby wątków maleje. Przyczyną tego stanu wyników prawdopodobnie jest zaimplementowanie w programie mechanizmów znacznie spowalniających działanie programu.