Hubert Kunikowski, Piotr Rzewnicki

Sprawozdanie z projektu

Równoległe rozwiązywanie problemu n-ciał

1. Wstęp

Celem projektu było zaimplementowanie równoległego rozwiązywania problemu n-ciał z wykorzystaniem OpenMP. Należało także dokonać analizy porównawczej programu równoległego z programem sekwencyjnym.

Problem n-ciał to zagadnienie mechaniki klasycznej polegające na wyznaczeniu toru ruchów wszystkich ciał danego układu n-ciał o danych masach, prędkościach i położeniach początkowych, w oparciu o prawa ruchu o założenie, że ciała oddziałują ze sobą zgodnie z prawem grawitacji Newtona.

Rozwiązanie problemu dla kilku ciał nie stanowi problemu podczas wykorzystywania programowania sekwencyjnego. W przypadku rozpatrywania problemu dotyczącego kilkuset bądź kilku tysięcy ciał, mnogość obliczeń może powodować problemy. Czas obliczeń wydłuża się znacznie, gdy dodajemy kolejne ciała. Dzieje się tak gdyż złożoność obliczeniowa takiego problemu to $O(n^2)$, gdyż każde z ciał oddziałuje z każdym pozostałym ciałem.

Każde ciało, do wyliczenia swojej następnej pozycji, potrzebuje wyłącznie aktualnego położenia innych ciał, gdyż z nimi oddziałuje. Ten fakt umożliwia zrównoleglenie obliczeń.

2. Uruchomienie programu

2. 1 Uruchomienie programu głównego

Program należy skompliować za pomocą polecenia:
cc -fopenmp main.c -lm

Następnie należy uruchomić program przy pomocy polecenia:
./a.out %{file} %{thread_num}

, gdzie jako pierwszy argument (file) podajemy nazwę pliku z danymi wejściowymi. Drugim argumentem jest liczba wątków. W przypadku braku podania tych informacji, program przyjmie wartości domyślne („data.txt”, 2).

W wyniku działania programu utworzy się plik „output.txt” zawierający obliczone kolejne pozycje ciał.

2. 2 Uruchomienie generatora losowych danych

Napisaliśmy również program w języku Python, który generuje losowe dane. Program można uruchomić poleceniem:
python3 input_data_creator.py ${num}

, gdzie zmienna num oznacza liczbę ciał dla których dane mają zostać utworzone. W wyniku działania programu powstanie plik „data.txt”, który można wczytać.

2. 3 Dane wejściowe

Jako dane wejściowe należy podać plik. Pierwszy wiersz pliku to liczba ciał, a kolejny to liczba parametrów (8), którymi opisane jest każde ciało. Każdy kolejny wiersz będzie zawierał informację o kolejnych ciałach. Każde ciało opisane jest kolejno następującymi parametrami oddzielonymi spacją:
- nazwa/id,
- masa.
- pozycja X,
- pozycja Y.
- pozycja Z,
- prędkość X,
- prędkość Y,
- prędkość Z.

Przykład danych wejściowych:

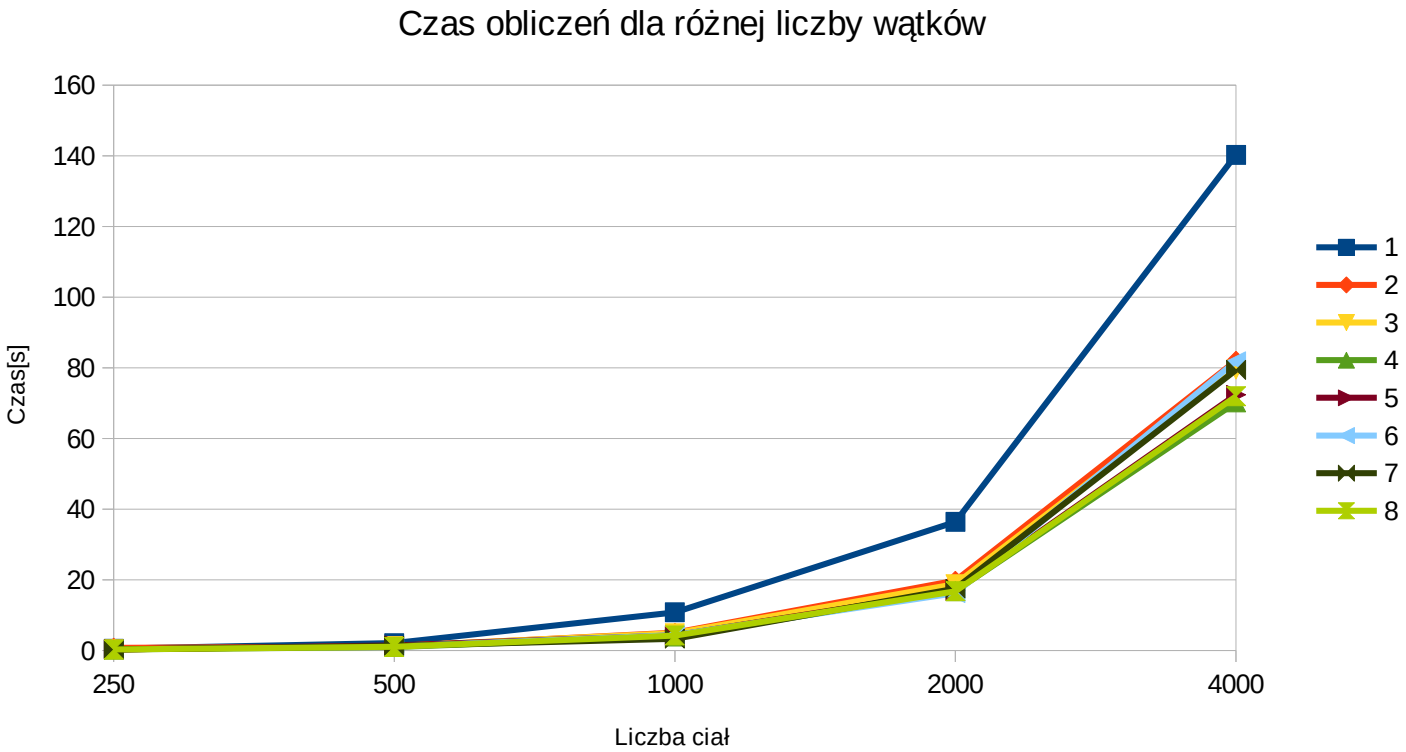
3
8
0 6413136262 78751 -54961 -45187 5715201423 4834083257 2732034528
1 7073526193 67144 -81698 -97442 -4390918287 2741656642 -494145145
2 119092962 -61892 38192 84803 4066456158 7012675200 664917222

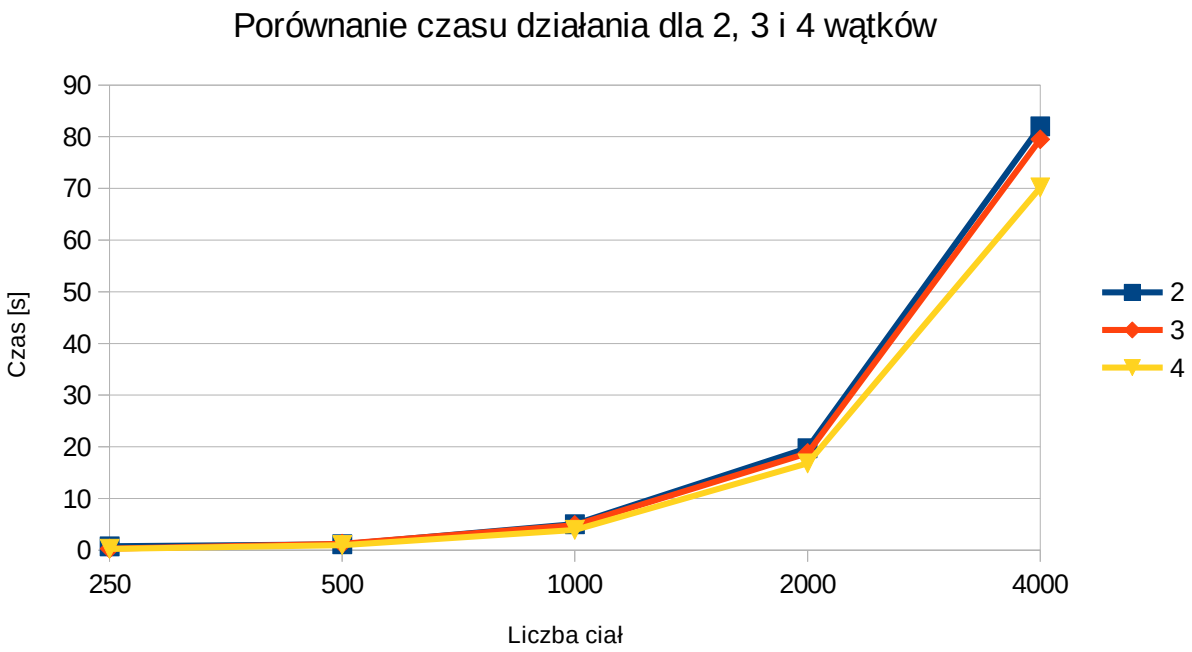
3. Uzyskane wyniki

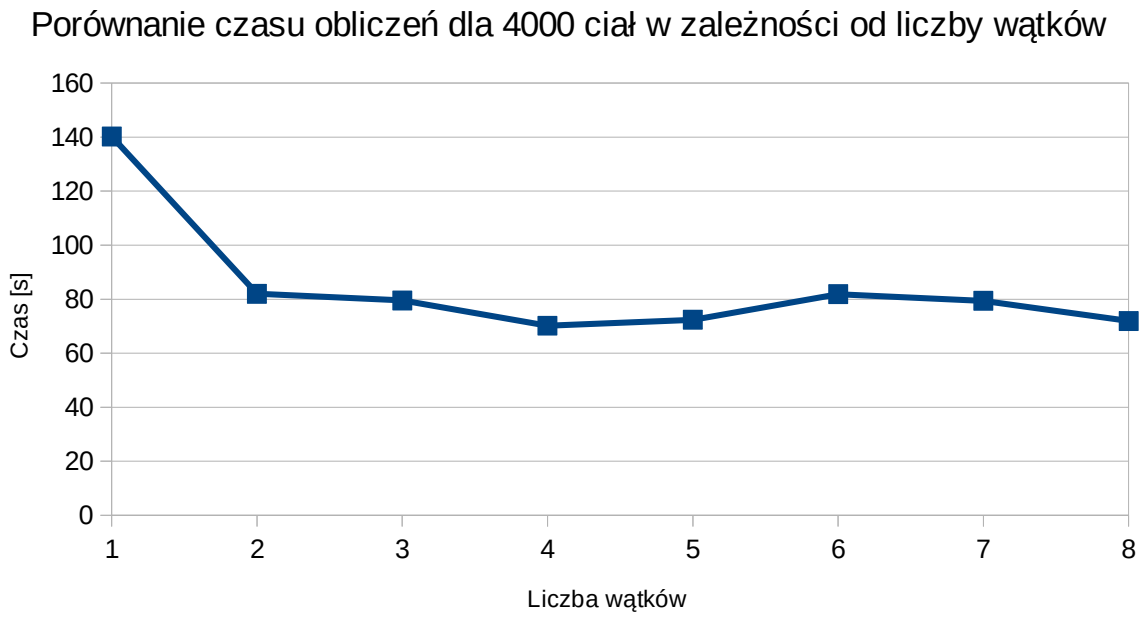
Analizę działania programu dokonywaliśmy dla 100 iteracji, a różnica czasowa, użyta w obliczeniach, pomiędzy kolejnymi iteracjami wynosiła 1000 sekund.

Czas obliczeń dla liczby wątków z zakresu 1-8 dla różnej liczby ciał (czas podano w sekundach):

Liczba wątków	**Liczba ciał**				
	250	**500**	**1000**	**2000**	**4000**
1	0,52	2,08	10,81	36,39	140,25
2	0,75	1,13	5,02	19,74	82,01
3	0,32	1,25	4,86	18,79	79,51
4	0,27	0,99	3,92	16,80	70,17
5	0,32	1,45	4,29	16,91	72,36
6	0.29	1,13	4,36	16,23	81,87
7	0,29	1,29	4,34	17,5	79,38
8	0,28	1,01	4,25	16,75	71,93




Porównanie czasu działania dla 2, 3 i 4 wątków
90
80
70
60
50
40
30
20
10
0
Czas [s]
250
500
1000
2000
4000
Liczba ciał
2
3
4


Porównanie czasu obliczeń dla 4000 ciał w zależności od liczby wątków
160
140
120
100
80
60
40
20
0
Czas [s]
1
2
3
4
5
6
7
8
Liczba wątków

Uzyskane przyspieszenie względem obliczeń sekwencyjnych:

	Liczba ciał				
Liczba Wątków	250	500	1000	2000	4000
2	0,69	1,84	2,15	1,84	1,71
3	1,63	1,66	2,22	1,94	1,76
4	1,93	2,1	2,76	2,17	2
5	1,63	1,43	2,52	2,15	1,94
6	1,79	1,84	2,48	2,24	1,71
7	1,79	1,61	2,49	2,08	1,77
8	1,86	2,06	2,54	2,17	1,95

Efektywność zrównoleglenia programu, a więc iloraz przyspieszenia i ilości wątków prezentuje się następująco:

	Liczba ciał				
Liczba Wątków	250	500	1000	2000	4000
2	0,35	0,92	1,08	0,92	0,86
3	0,54	0,55	0,74	0,65	0,59
4	0,48	0,53	0,69	0,54	0,5
5	0,33	0,29	0,5	0,43	0,39
6	0,3	0,31	0,41	0,37	0,29
7	0,26	0,23	0,36	0,3	0,25
8	0,23	0,26	0,32	0,27	0,24

4. Wnioski

Zgodnie z oczekiwaniami rozwiązanie tego problemu w sposób równoległy było dużo szybsze niż w sposób sekwencyjny. Największe przyspieszenie zostało osiągnięte dla 4 i 8 wątków. Jest to spowodowane, tym że procesor, w komputerze na którym przeprowadzaliśmy testy, jest 4-wątkowy.

W wyniku zrównoleglenia udało się uzyskać nawet ponad 2,5 – krotne przyspieszenie. Zauważyć można, że wraz ze zwiększającą się liczbą wątków, czas wykonywania programu skraca się, by później wzrosnąć. Jest to spowodowane „wąskim gardłem” tej metody, a więc architekturą procesora.