Multi-core Architectures 2020/2021
Parallel implementation evaluation report

1. Project details

Project title:	Problem N-ciał
Student's name:	Piotr Walkusz
Index number:	165231
Email:	piotrwalkusz1@gmail.com

2. Description of data used for experiments (including examples, when possible)

Na wejściu program otrzymuje listę ciał, gdzie każde ciało jest opisywane po przez pozycję, prędkość początkową i masę. W eksperymentach zakładamy, że wszystkie ciała mają taką samą masę i zerową prędkość początkową. Pozycja ciał w każdej z osi (x i y) jest losowana z zakresu (0.25, 0.75).

3. Environment #1 description

Ubuntu 18.04.4 LTS
Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
OpenMP 4.5

4. Environment #2 description

Ubuntu 18.04.4 LTS
Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
GeForce GTX 1060 6GB
CUDA 10.1

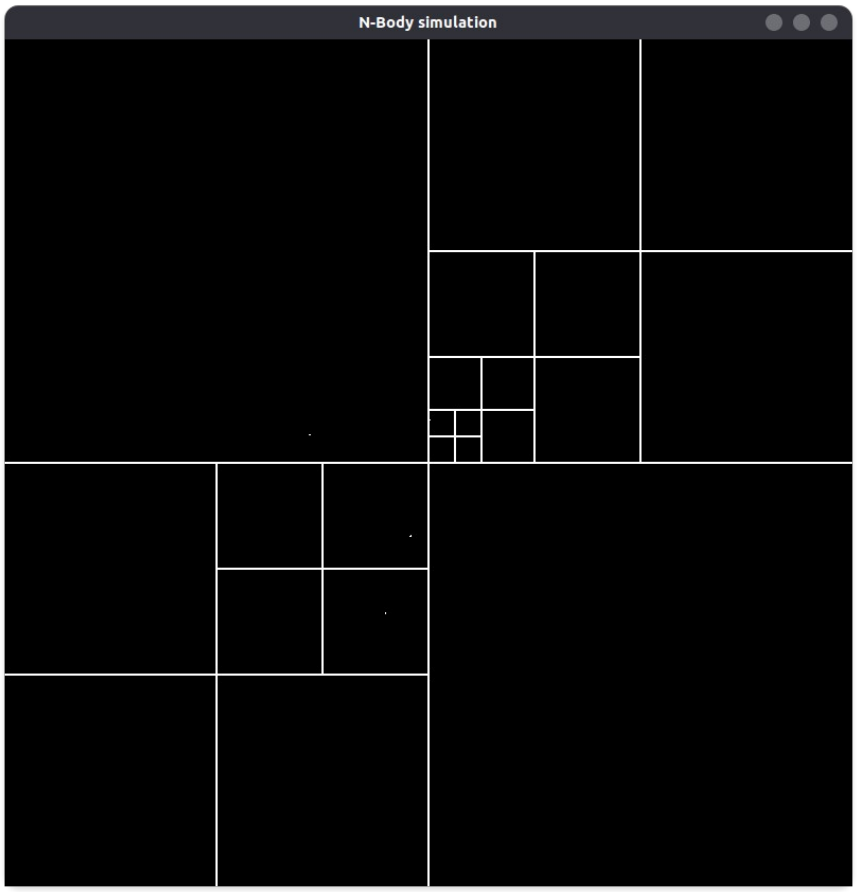
5. Test results

Implementation	**Execution time***	
	Mean [s]	**Uncertainty [s]**
OpenMP	38.835	0.2317
CUDA	3.924	0.0359

* calculated over 10 executions, uncertainty calculated as: $\Delta T = \frac{T_{max} - T_{min}}{2}$

6. Implementation #1 details

W implementacji OpenMP zastosowano algorytm Barnes-Hut. Każdą iterację można podzielić na trzy etapy. W pierwszym etapie budujemy drzewo czwórkowe, czyli takie drzewo, którego każdy wierzchołek reprezentuje przestrzeń dwuwymiarową, a jego dzieci (maksymalnie 4) dzielą tą przestrzeń na 4 mniejsze podprzestrzenie. Dodajemy kolejne ciała do takiego drzewa używając pozycji ciała jako klucza. Jeżeli w jakieś przestrzeni znajduje się już ciało, dzielimy tą przestrzeń na podprzestrzenie. Przykładowy podział (białe kropi to ciała):



W drugim etapie liczymy środek ciężkości i sumę mas wszystkich ciał w każdej z podprzestrzeni. W trzeci, ostatnim etapie dla każdego ciała liczymy siłę z jaką oddziałują na nie inne ciała wykorzystując fakt, że oddziaływanie dużej grupy oddalonych ciał możemy przybliżyć przez wykorzystanie jedynie środka ciężkości i sumy mas tych ciał.

Zrównoleglenie dodawania ciał do drzewa czwórkowego uzyskano dzięki mechanizmowi locków (omp_set_lock). Obliczanie środka ciężkości i masy dla wierzchołków drzewa czwórkowego wykonano za pomocą metody dziel i rządź. Wątki liczyły środek ciężkości i masę dla każdego z 4 podprzedziałów rodzica, a następnie sumowano to u rodzica. Zrównoleglenie obliczania siły grawitacji dla każdego ciała już po zbudowaniu drzewa czwórkowego było banalne, ponieważ na tym etapie następuje wyłącznie odczyt z drzewa, więc siła grawitacji dla każdego ciała może być liczona niezależnie.

7. Implementation #2 details

W implementacji CUDA wykorzystano algorytm Particle-Particle, czyli liczymy siłę grawitacji pomiędzy każdą parą ciał. Dla uproszczenia implementacji zakładamy, że liczba ciał musi być podzielna przez wielkość bloku, a wielkość bloku musi być podzielna przez 4. Blok o wielkości D oblicza przyspieszenie, zmianę prędkości i zmianę położenia dla D ciał. Aby nie pobierać pozycji i masy innych ciał w każdym bloku z pamięci globalnej, każdy blok pobiera do pamięci shared informację o 2D ciał. Pozycja x i y jest zapisywana razem z masą w zmiennej typu float3, aby zastosować wektoryzację w kopiowaniu wartości. Dane do GPU są przesyłane tylko raz, później przy każdym wywołaniu kernela tylko zamieniamy ze sobą wskaźniki do listy starych i nowych pozycji. Kompilujemy z flagą use_fast_math, która czterokrotnie przyspieszyła działanie programu.

8. Survey

Fill the answers for questions related to frameworks that you used in your project:

a. How many lines of code did you write for:
    i. OpenMP implementation: 230
    ii. CUDA implementation: 150
    iii. MPI implementation: -

b. How would you describe programming difficulty of each framework/interface in 1-10 scale (1 – easy, 10 – difficult):
    i. OpenMP: 7
    ii. CUDA: 4
    iii. MPI: 10