

Lab WSP 2: Całkowanie numeryczne

Opracował: Artur Chrzanowski

Na 2. ćwiczeniach laboratoryjnych należy napisać program, który oblicza całkę pod polem stworzonym ze zbioru punktów. Następnie należy dokonać porównania szybkości realizacji zadania dla rozwiązań CPU i GPU. Limity czasowe na oddanie zadania przedstawione są w zadaniach na platformie e-nauczanie. Przekroczenie limitu czasowego może skutkować obniżeniem oceny.

1 Opis zadania

Zadanie składa się z kilku podetapów. Najpierw należy napisać generator danych w postaci funkcji. Funkcja powinna przyjmować parametr informujący ile elementów należy wygenerować. Format danych wyjściowych jest dowolny (można stworzyć tablicę/listę/wskaźnik punktów-2D/wartości X i Y itd.). Punkt składa się ze współrzędnych X i Y. Wartości punktów X i Y powinny być nieujemne (zawierać się w pierwszej ćwiartce układu prostokątnego) i mniejsze od 100 000. Punkty nie powinny powtarzać się względem osi X. Pole figury poddanej całkowaniu jest tworzone przy użyciu krzywej, stworzonej przez połączenie punktów X i Y interpolacją liniową (prosty odcinek) najbliższych punktów względem współrzędnej X oraz osi X układu. Pole figury należy wyliczyć za pomocą całkowania numerycznego. Generator należy napisać dla CPU. Współrzędne X i Y są zmiennoprzecinkowe.

Wartości punktów należy generować losowo. Z funkcji należy uzyskać nieposortowany zbiór punktów względem osi X. Następnie należy napisać kod dla CPU sortujący punkty i obliczający całkę numeryczną z pola figury tworzonej przez zbiór punktów. Powtórzyć kroki dla GPU. Należy wybrać dowolną metodę numeryczną (preferowane jest użycie bardziej zaawansowanych metod pozwalających na minimalizację błędów.)

2 Zadania do wykonania

Napisać program na CPU, realizujący sortowanie oraz całkowanie numeryczne i zwracający wartość pola wygenerowanego przez zbiór punktów.

1. Następnie należy rozszerzyć program, tak aby te samo zadanie wykonane zostało na GPU. Wyniki programu GPU i CPU powinny być identyczne. Finalnie, należy podać przyspieszenie GPU względem CPU
2. Opisać otrzymane rezultaty w formie sprawozdania i umieścić sprawozdanie w folderze z kodem źródłowym. Skompresowany folder należy przesłać do zadania 2. na platformie e-nauczaniu.

3 Sprawozdanie

Elementy, które muszą być zawarte w sprawozdaniu i kodzie:

- Napisany projekt musi kompilować się bezbłędnie na kartach graficznych Nvidia z CUDA toolkit o wersji ≤ 8.0 . Program nie może wymagać więcej pamięci do obliczeń niż może udostępnić mu urządzenie GPU. Należy opracować plik makefile, który przeprowadzi kompilację po wywołaniu komendy make dla systemów Linux lub projekt Visual Studio (2017/2019) dla systemów Windows. W przypadku nie kompilującego się lub nie działającego programu mogą zostać nadane kary punktowe. W przypadku użycia dodatkowych bibliotek należy je dołączyć do projektu.
- Sprawozdanie powinno zawierać informacje takie jak wykres przedstawiający przyspieszenie oferowane przez GPU w stosunku dla CPU, przy zmiennej ilości generowanych punktów. Należy podać także typ karty graficznej na jakiej test był wykonywany.

- W sprawozdaniu powinna być informacja, jakie trudności napotkał student, pisząc samodzielnie program oraz czego się nauczył.
- W sprawozdaniu powinna być informacja, jakie nowości student opracować i zaimplementować pozwalające prowadzącemu zwiększyć ocenę ponad standardowe 8 punktów. Przykładowo można zadanie zrealizować stosując pamięć współdzieloną, porównać szybkość i precyzję z użyciem liczb zmiennoprzecinkowych pojedynczej i podwójnej precyzji lub wykorzystać asynchroniczne przesyłanie danych. Mile widziana charakterystyka złożoności obliczeniowej.
- Programy i sprawozdania będą sprawdzane pod kątem samodzielności pisania. Zgłoszenie do rozliczenia kopii rozwiązania od innego studenta lub z serwisów internetowych skutkuje brakiem zaliczenia przedmiotu.
- Do Państwa dyspozycji zazwyczaj jestem dostępny w poniedziałki w godzinach 9 - 17:00. Z uwagi na zawirowania spowodowane koronawirusem w sprawie konsultacji proszę zgłaszać się na maila podanego w kursie e-nauczanie. Konsultacje odbywać mogą się w formie zdalnej.

4 Kryteria Oceny

Za poprawnie wykonanie wszystkich elementów zadania można uzyskać 8 punktów na dziesięć możliwych. Punkty dodatkowe (+4) można uzyskać za ponadprzeciętną jakość kodu i sprawozdania. Punkty można stracić przy pominięciu sortowania (-2), uzyskaniu względnie słabych rezultatów przy implementacji GPU, nieschludnym sprawozdaniu i błędach w działaniu programu lub generacji danych.