

Projekt na zaliczenie

1 Podstawowe założenia

Napisać program ilustrujący zastosowanie GPU dla przyspieszenia obliczeń dotyczących wybranego zagadnienia. Przykładowa lista zagadnień znajduje się w dalszej części.

Założenia ogólne:

- język implementacji dowolny, jednak kod dla *kerneli* powinien być podany jawnie, a nie generowany automatycznie przez dodatkową bibliotekę;
- można zastosować platformę CUDA albo OpenCL;
- program powinien zawierać implementację sekwencyjną rozwiązania, tj. przeznaczoną dla CPU oraz drugą – równoległą – dla GPU;
- implementacja powinna oferować jak największą wydajność, ale bez utraty ogólności, np. przetwarzania tylko danych wejściowych o wymiarach będących potęgą liczby 2;
- równoległa implementacja powinna zapewniać rzeczywiste skrócenie czasu obliczeń w odniesieniu do wersji sekwencyjnej – należy to wziąć pod uwagę wybierając typ rozwiązywanego problemu;
- nie ma konieczności zrównoleglania „całego” algorytmu. Wystarczy ten fragment, który odpowiada za większość czasu obliczeń, tak by możliwe było uzyskanie przyspieszenia;
- dla każdej wykonywanej operacji program powinien wyświetlać czas wykonania, w przypadku długotrwałych obliczeń można wyświetlić pasek postępu;
- interfejs graficzny programu nie jest wymagany, jednak należy udostępnić interfejs tekstowy umożliwiający ustalanie danych wejściowych i wartości parametrów algorytmów;
- program może pisać indywidualnie, bądź w parach – proszę jednak uwzględnić, że poziom trudności w przypadku projektów realizowanych w parach powinien być odpowiednio wyższy niż w przypadku projektów realizowanych indywidualnie.

2 Przykładowe tematy

1. Program do przetwarzania obrazów (fotografii) za pomocą operacji, takich jak rozmycie, wyostrzanie, wyrównanie histogramu. Należy zrównoleglić również takie operacje, jak znajdowanie minimum / maksimum, wyznaczanie sum prefiksowych, sortowanie.
2. Program realizujący wybrany algorytm metaheurystyczny do optymalizacji kombinatorycznej, taki jak algorytm genetyczny, algorytm mrówiskowy, optymalizacja stadna cząstek (PSO) itp. Jako rozwiązywany problem należy wybrać problem należący do klasy NP-zupełnej, np. TSP. Można również wybrać problem optymalizacji ciągłej.

3. Program do symulacji fizycznej, np. ruchu cząsteczek oddziałujących na siebie wzajemnie siłą grawitacji, elektrostatyczną itp.
4. Program implementujący podstawowy algorytm renderowania grafiki trójwymiarowej metodą śledzenia promieni (ang. ray tracing). Na potrzeby projektu generowane sceny powinny składać się tylko ze sfer oraz płaszczyzn. Również przyjęty model cieniowania może być bardzo prosty, np. Phonga.

Przykładowe dokumenty wprowadzające do tematyki:

- <http://www.ics.uci.edu/~gopi/CS211B/RayTracing>
- <https://www.cs.unc.edu/~rademach/xroads-RT/RTarticle.html>

3 Uwagi

Podstawą pozytywnej oceny jest poprawna implementacja programu i realizacja przyjętych założeń. Pod uwagę brane są również złożoność podjętego zagadnienia oraz efektywność obliczeniowa uzyskanej implementacji.