

Problemy do rozwiązania na zaliczenie drugiej części wykładu z programowania równoległego, semestr zimowy 2012/13 – CUDA

1. Zbadaj, jak prędkość mnożenia macierzy kwadratowych  $N \times N$  zależy od ich rozmiaru  $N$ . W tym celu wykonaj wykresy:
  - czasu jednego mnożenia,  $t(N)$ , w funkcji  $N$
  - $t(N)/N^2$  w funkcji  $N$
  - wykonaj analogiczne symulacje na swoim CPU, porównaj otrzymane wyniki (być może na tych samych rysunkach), wyciągnij wnioski.

Uwaga: do wykonania symulacji na GPU użyj biblioteki cuBLAS. Obliczenia wykonuj w arytmetyce zmiennoprzecinkowej podwójnej precyzji.
2. Napisz program, który na GPU rozwiąże następujący problem: dane są dwa ciągi  $(x_i)$ ,  $(y_i)$ ,  $i = 0, \dots, N-1$ , definiujące położenia  $N$  punktów na płaszczyźnie. Kolejne trójki tych punktów definiują położenia wierzchołków  $N-2$  trójkątów. Innymi słowy, wierzchołki pierwszego trójkąta mają indeksy  $0, 1, 2$ ; wierzchołki drugiego trójkąta – indeksy  $1, 2, 3$  itd.
  - Napisz program, który w osobnej tablicy zapisze pola powierzchni tych trójkątów.
  - Przetestuj efektywność swojego programu dla kilku wartości  $N$ .
  - Porównaj jego wydajność z wydajnością analogicznego programu rozwiązującego ten problem na CPU.
3. Dane są 3 ciągi liczb  $(v_{x_i})$ ,  $(v_{y_i})$ ,  $(m_i)$ ,  $i = 0, \dots, N-1$ , definiujące składowe  $x$ ,  $y$  prędkości punktów na płaszczyźnie oraz ich masy.
  - Napisz program w CUDA wyznaczający średnią energię kinetyczną cząstek tego układu.
  - Zbadaj szybkość działania tego programu dla kilku wartości  $N$ .
  - Porównaj efektywność Twojego rozwiązania z kodem na CPU.
4. Napisz kernel CUDA, który kopiuje jedną tablicę danych typu double do drugiej tablicy. Za pomocą tego kernela:
  - Zbadaj, w jaki sposób prędkość kopiowania danych w GPU zależy od wielkości bloku.
  - Porównaj uzyskane przez siebie wyniki z teoretyczną przepustowością pamięci twojego urządzenia.
  - Porównaj uzyskane przez siebie wyniki z wartością uzyskaną w programie bandwidthTest z CUDA SDK.
5. Napisz program, który na GPU rozwiąże następujący problem: dany jest ciąg par punktów  $(x_i, y_i)$ ,  $i = 0, \dots, N-1$ , definiujących punkty na płaszczyźnie, oraz kąt  $\alpha$ .
  - Napisz program, który przekształci strumień wejściowy w nowy strumień, zawierający współrzędne punktów ze strumienia wejściowego obróconych o kąt  $\alpha$  względem środka układu współrzędnych.
  - Przetestuj jego efektywność dla kilku wartości  $N$ .
  - Porównaj jego wydajność z wydajnością analogicznego programu rozwiązującego ten problem na CPU.
6. Napisz program, który na GPU rozwiąże następujący problem: dany jest ciąg czwórek punktów  $(x_i, y_i, v_{x_i}, v_{y_i})$ ,  $i = 0, \dots, N-1$ , definiujących punkty na płaszczyźnie oraz pewna liczba  $dt$ .
  - Napisz program, który zamieni  $x_i$  na  $x_i + dt \cdot v_{x_i}$  oraz  $y_i$  na  $y_i + dt \cdot v_{y_i}$  dla każdego  $i$ .
  - Przetestuj jego efektywność dla kilku wartości  $N$ .
  - Porównaj jego wydajność z wydajnością analogicznego programu rozwiązującego ten problem na CPU.

Uwaga: czwórkę liczb zmiennopozycyjnych wygodnie zapisuje się w typie danych float4.
7. Napisz program, który na GPU rozwiązuje następujący problem: dany jest ciąg czwórek liczb  $(a_i, r_i, g_i, b_i)$ ,  $i = 0, \dots, N-1$ , o wartościach typu float z przedziału  $[0, 1]$ , definiujących kolory przypisane kolejnym pikselom obrazu.
  - Napisz program, który zamieni  $r_i, g_i, b_i$  na ich średnią arytmetyczną.
  - Przetestuj efektywność tego programu dla kilku wartości  $N$ .

- Porównaj jego wydajność z wydajnością analogicznego programu rozwiązującego ten problem na CPU.

Uwaga: czwórkę liczb zmiennopozycyjnych wygodnie zapisuje się w typie danych float4;

8. Napisz program, który na GPU rozwiąże następujący problem: dany jest ciąg par punktów  $(x_i, y_i)$ ,  $i = 0, \dots, N-1$ , definiujących punkty na płaszczyźnie oraz N-elementowa tablica n przekazana poprzez wskaźnik do pierwszego elementu.

- Napisz program, który w elemencie i tablicy n,  $i = 0, \dots, N-1$ , zapisze indeks k takiego punktu  $(x_k, y_k)$ , że żaden inny punkt, poza  $(x_i, y_i)$ , nie leży bliżej punktu  $(x_i, y_i)$  w sensie metryki Euklidesowej.
- Przetestuj efektywność swojego programu dla kilku wartości N.
- Porównaj wydajność swojego rozwiązania z wydajnością analogicznego programu rozwiązującego ten problem na CPU.

Uwaga: Oddziaływanie „każdy z każdym” to wdzięczny temat do optymalizacji programu równoległego, ale w tym wypadku wystarczy rozwiązanie po prostu działające. W CUDA pary liczb zmiennopozycyjnych wygodnie zapisuje się w typie float2 lub double2.

9. Napisz program, który na GPU rozwiąże następujący problem: dany jest ciąg trójek liczb  $(x_i, y_i, z_i)$ ,  $i = 0, \dots, N-1$ , definiujących współrzędne środków gwiazd w przestrzeni oraz N-elementowa tablica v przekazana poprzez wskaźnik do pierwszego elementu. W środku układu współrzędnych znajduje się Bardo Masywny Obiekt i interesują nas energie potencjalne gwiazd w polu grawitacyjnym Obiektu.

- Napisz program, który w elemencie i tablicy v,  $i = 0, \dots, N-1$ , zapisze odwrotność odległości punktu  $(x_i, y_i, z_i)$  od środka układu współrzędnych.
- Przetestuj efektywność swojego programu dla kilku wartości N.
- Porównaj wydajność swojego rozwiązania z wydajnością analogicznego programu rozwiązującego ten problem na CPU.

Uwaga: Trójki liczb zmiennopozycyjnych wygodnie zapisuje się w typie float3 lub double3. Możesz też użyć typów float4 i double4 i założyć, że potencjały zapisuje się w czwartej składowej tych struktur zamiast w tablicy v.

10. Podobno wyznaczanie reszty z dzielenia na GPU jest operacją kosztowną. Sprawdź to. Zakładając, że masz ciąg N liczb całkowitych (np. kolejnych liczb  $1, \dots, N$  lub liczb wygenerowanych losowo):

- Napisz program, który w innym strumieniu danych (tj. w osobnej tablicy) zapisze reszty z dzielenia liczb z tablicy wejściowej przez 11.
- Przetestuj efektywność swojego programu dla kilku wartości N.
- Porównaj wydajność swojego rozwiązania z wydajnością analogicznego programu rozwiązującego ten problem na CPU.
- Jak wykonanie programów na CPU i GPU zmieni się, jeżeli zamiast wyznaczać resztę z dzielenia przez 11 Twój program będzie w strumieniu wyjściowym zapisywał reszty z dzielenia przez 8?

Uwaga:

Proszę zgłaszać mi numery wybranego zadania. Jedno zadanie może zostać przydzielone maksymalnie 2 osobom. Projekty rozwiązujemy indywidualnie. Zaliczenie projektu wymaga przedstawienia wyników swoich obliczeń w formie pisemnej, w formie krótkiego raportu, np. składającego się z krótkiego wstępu (opis problemu), uzyskanych wyników (rysunki), ich omówienia i wniosków. Samo omówienie programu zostawiamy na egzamin (ustny). W razie wątpliwości proszę o kontakt mailowy lub podczas zajęć.

**Rezerwacje zadań w semestrze zimowym 2012/13, stan na 3.01.2013:**

Zad 1: Iwona Opałka Rafał Lachtera Piotr Mazur

Zad 2:

Zad 3:

Zad 4: Krzysztof Cach

Zad 5: Piotr Olkiewicz

Zad 6: Paweł Franczak

Zad 7:

Zad 8:

Zad 9: Michał Naskręt; Michał Sokołowski

Zad 10: