Michał Szczygieł	Aleksander Śmierciak	18/06/2014

Zadanie 5 - Mnożenie macierzy na GPU

Programem spełniającym polecenie zadania 5. jest taki, który wykona mnożenie dwóch macierzy kwadratowych przy użyciu karty graficznej, np. za pomocą CUDA.

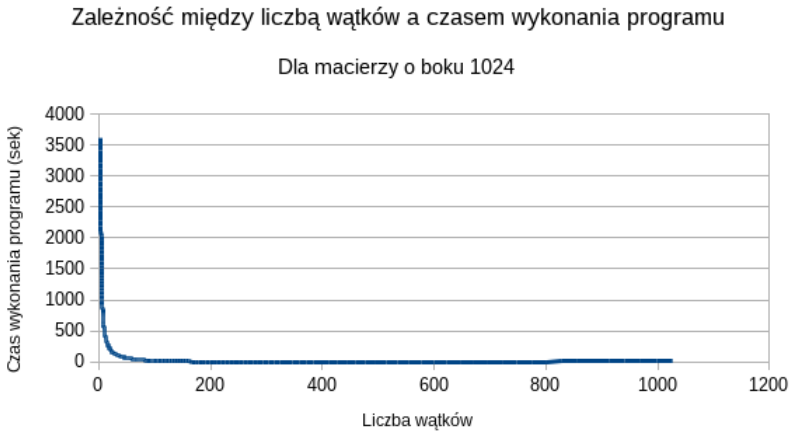
```
dim3 block(threadCount);
dim3 grid(matrixSize / block.x, matrixSize / block.y);
startTimer(start);
matrixMultiplyKernel<<<grid, block>>>(devA, devB, devC, matrixSize);
stopTimer(stop);
cout << readExecutionTimeInMillis(start, stop) << endl;
```

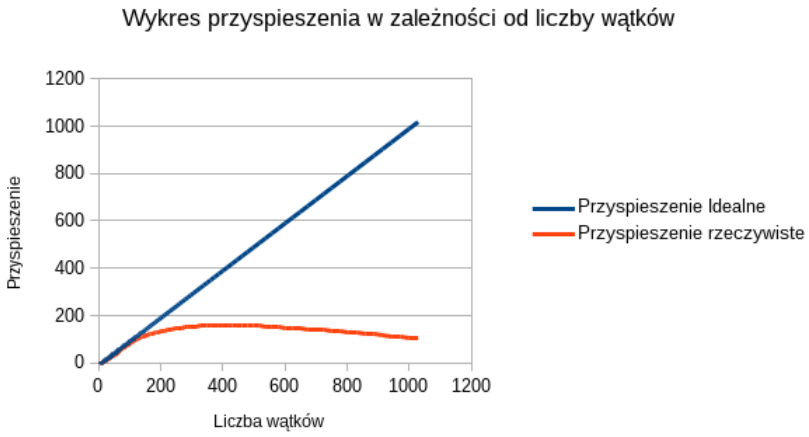
Etapy wykonania programu wykorzystującego CUDA można wydzielić następująco:

1. Ustawienie dwóch zmiennych typu struktury dim3: określającej wymiary macierzy i określającej wymiary pojedynczego bloku.
2. Zainicjalizowanie macierzy wejściowych, używając generatora liczb pseudo-losowych.
3. Zainicjalizowanie macierzy wejściowych przesłanych do jądra CUDA.
4. Transfer danych do macierzy jądra CUDA.
5. Wykonanie mnożenia macierzy, opatrzone zdarzeniami startu i stopu pomiaru czasu.
6. Zwolnienie pamięci.

Przebieg

Poprzednio utworzone wykresy kolumnowe zestawiające czas wykonania mnożenia macierzy CPU i mnożenia macierzy GPU zastąpiono wykresami prezentującymi przyrost wydajności wraz ze zwiększaną liczbą wątków.





Przy tworzeniu takiego programu nie wykonano żadnych optymalizacji; implementacja jest „naiwnym” mnożeniem odpowiadających komórek macierzy.
Dla macierzy kwadratowej o boku 1024 odnotowano czasy wykonania mnożenia rzędu 0.02 sekundy.