Metody Programowania Równoległego - CUDA

Maciej Mucha

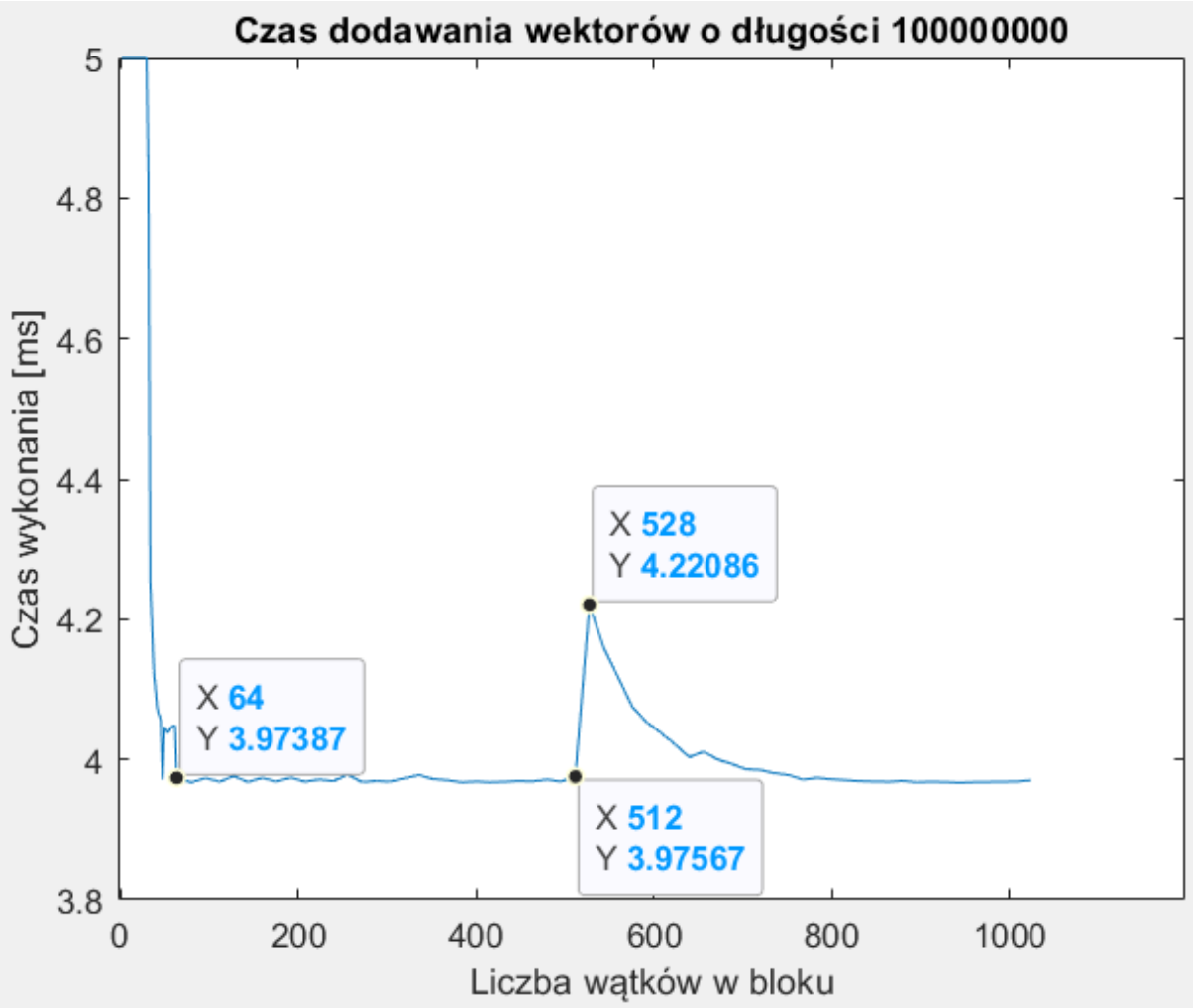
Wykorzystywany sprzęt

Testy były przeprowadzone na komputerze stacjonarnym wyposażonym w kartę graficzną Nvidia RTX 2060, wersja CUDA 11.6.

Wykonanie programu dostarczonego w przykładach CUDA, wypisującego charakterystykę systemu, zwróci następujące informacje:

```
 Device 0: "NVIDIA GeForce RTX 2060"
  CUDA Driver Version / Runtime Version          11.6 / 11.6
  CUDA Capability Major/Minor version number:    7.5
  Total amount of global memory:                 6144 MBytes (6442123264 bytes)
  (030) Multiprocessors, (064) CUDA Cores/MP:    1920 CUDA Cores
  GPU Max Clock rate:                            1755 MHz (1.75 GHz)
  Memory Clock rate:                             7001 Mhz
  Memory Bus Width:                              192-bit
  L2 Cache Size:                                 3145728 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(131072), 2D=(131072, 65536),
3D=(16384, 16384, 16384)
  Maximum Layered 1D Texture Size, (num) layers  1D=(32768), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(32768, 32768), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total shared memory per multiprocessor:        65536 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1024
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Model)
  Device supports Unified Addressing (UVA):      Yes
  Device supports Managed Memory:                Yes
  Device supports Compute Preemption:            Yes
  Supports Cooperative Kernel Launch:            Yes
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 43 / 0
```

Dodawanie wektorów

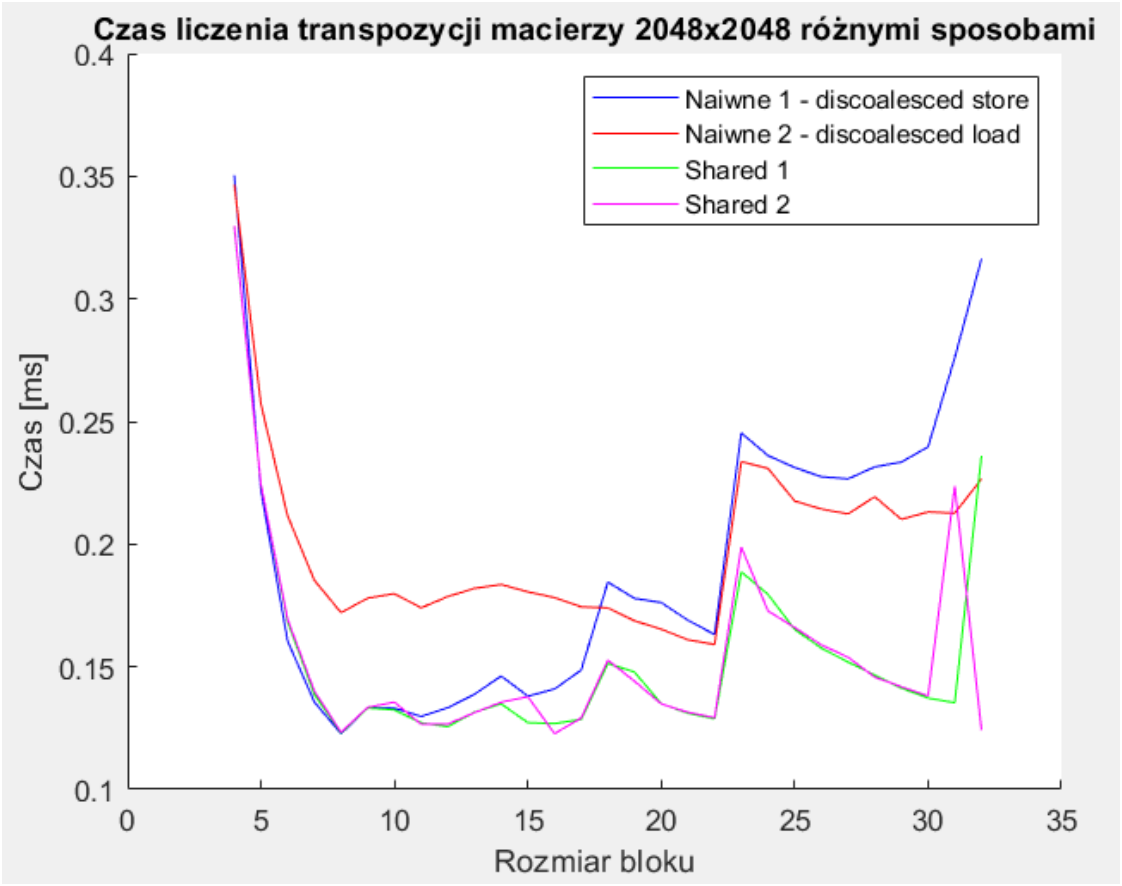


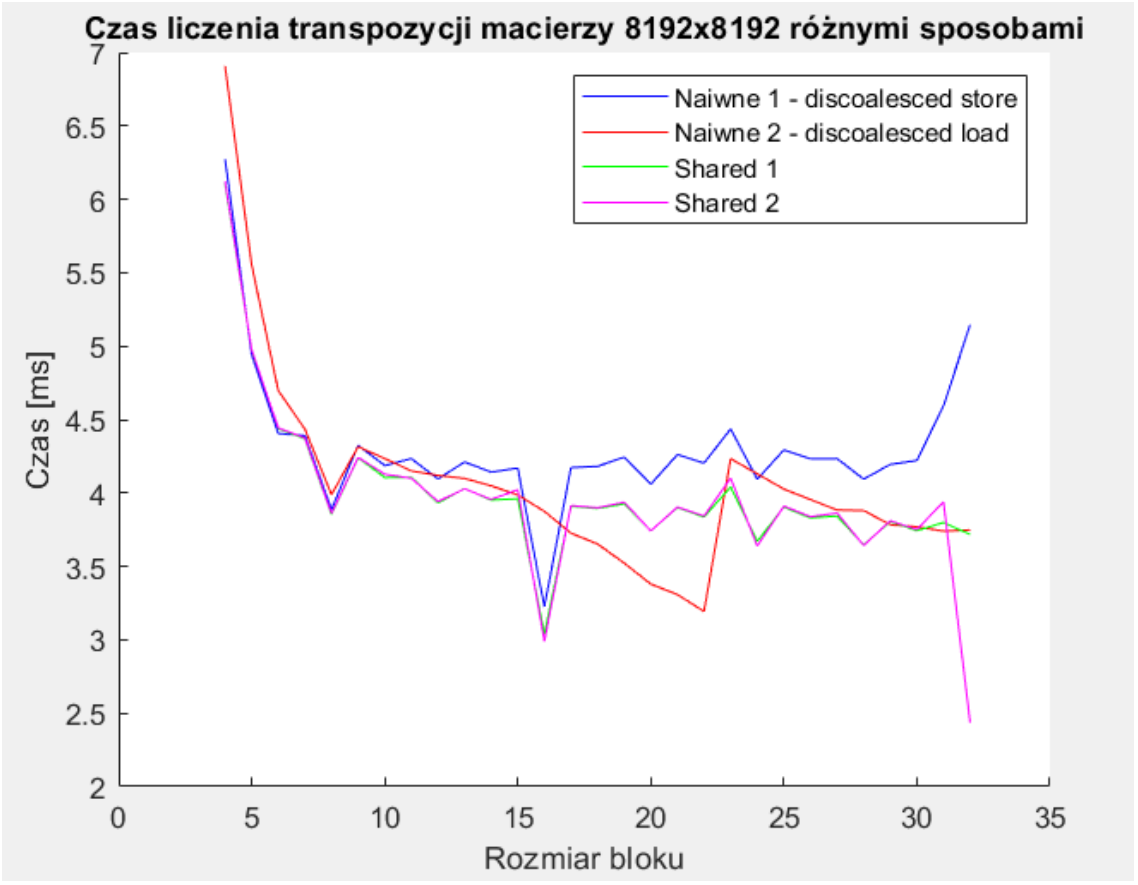
Początek, czyli dla liczby wątków mniejszej niż 64 osiągamy długie czasy wykonywania operacji dodawania wektorów. Następnie czas utrzymuje się na podobnym poziomie, aż do liczby wątków równej 512. Dla większych bloków znacznie wzrasta czas wykonania, który powoli spada do poprzedniej, najlepszej wartości.

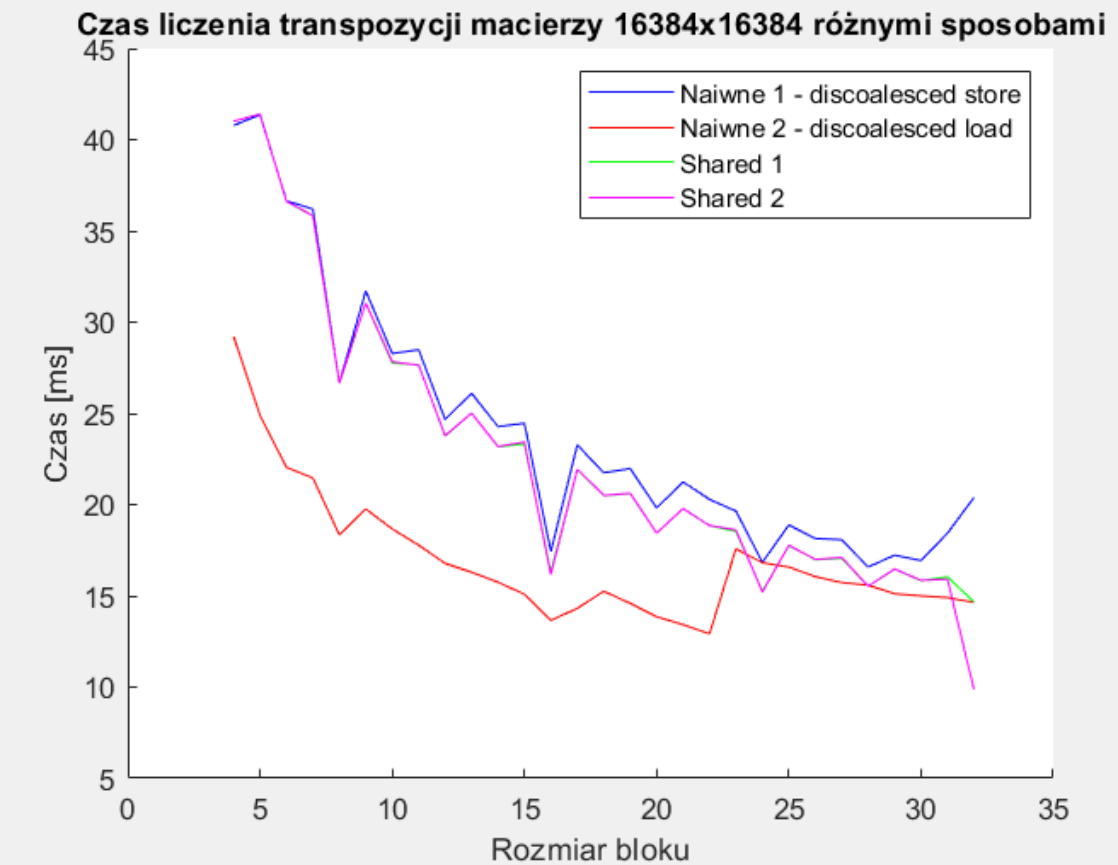
Transpozycja macierzy

Testowane były 3 rozmiary macierzy 2048x2048, 8192x8192 oraz 16384x16384, każdy w 4 różnych konfiguracjach:

1. Naiwne 1 bez użycia pamięci dzielonej z dostępem łączonym przy wczytywaniu, a nie przy zapisywaniu
2. Naiwne 2 bez użycia pamięci dzielonej z dostępem łączonym przy zapisywaniu, a nie przy wczytywaniu
3. Z pamięcią dzieloną w wersji 1
4. Z pamięcią dzieloną w wersji 2

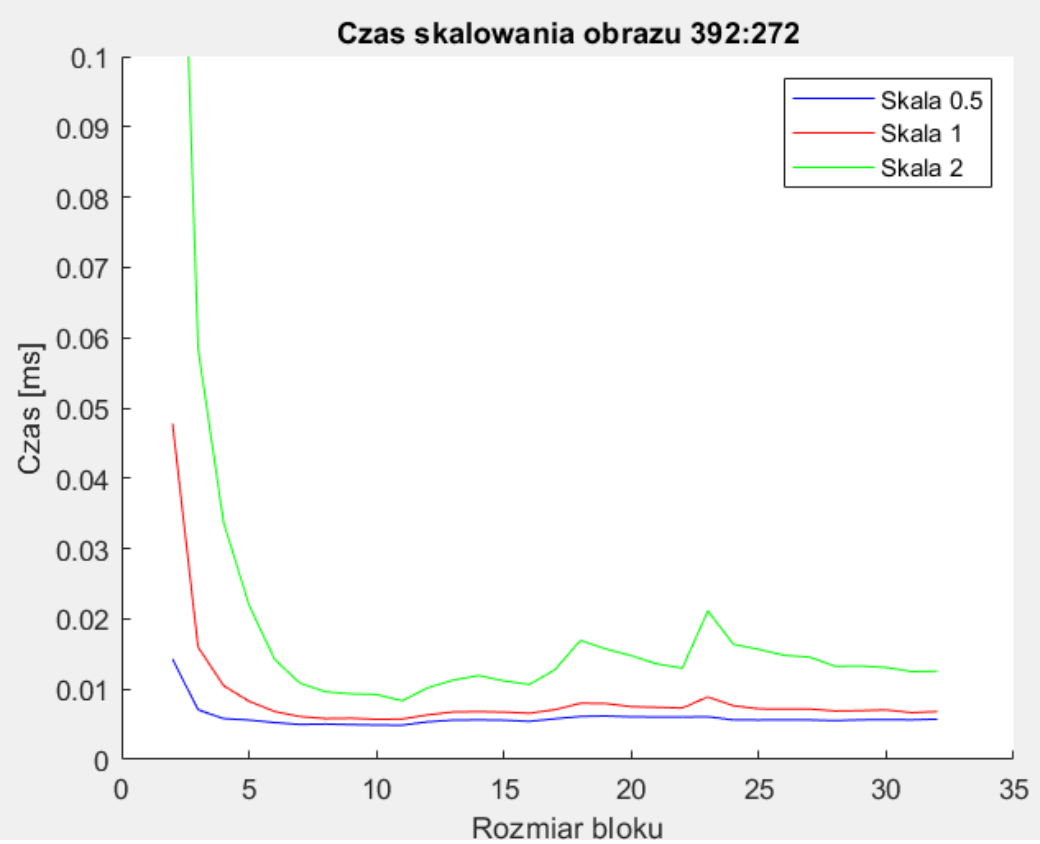

Czas liczenia transpozycji macierzy 2048x2048 różnymi sposobami
Naiwne 1 - discoalesced store
Naiwne 2 - discoalesced load
Shared 1
Shared 2
Czas [ms]
Rozmiar bloku
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0
5
10
15
20
25
30
35

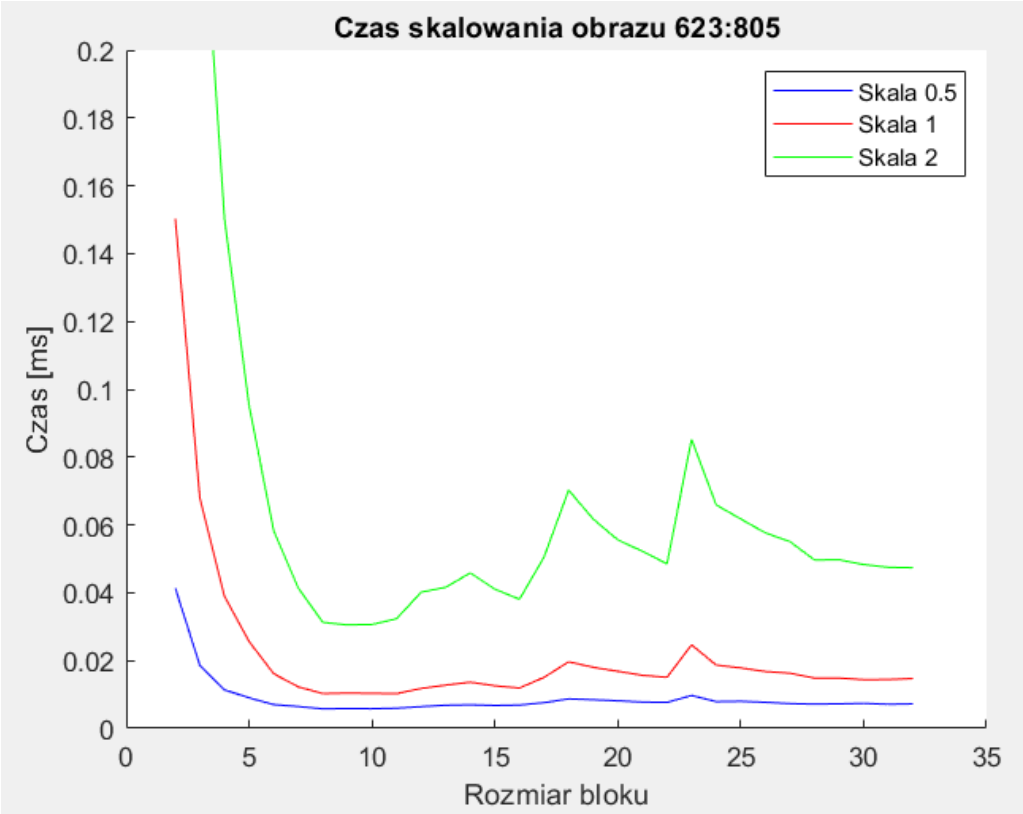

Czas liczenia transpozycji macierzy 8192x8192 różnymi sposobami
Naiwne 1 - discoalesced store
Naiwne 2 - discoalesced load
Shared 1
Shared 2
Czas [ms]
Rozmiar bloku
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
0
5
10
15
20
25
30
35



W najmniejszej macierzy wersje z pamięcią dzieloną były najszybsze dla wszystkich rozmiarów bloków, ale większych macierzy już nie zawsze.
Pierwszy znaczący spadek czasu wykonania wersji naiwnej 1, shared 1 i 2 był przy bloku rozmiarze 16 (256 wątków na blok).
Znaczący wzrost czasu dla wszystkich opcji macierzy 248 oraz opcji naiwnej 2 macierzy większych występuje przy rozmiarze 23 (529 wątków na blok) co pokrywa się ze spadkiem wydajności z testów dodawania wektorów.
Najlepsze wartości niezależnie od rozmiaru macierzy osiągnęła wersja 2 z pamięcią dzieloną dla rozmiaru bloku 32 (1024 wątki na blok)

Skalowania obrazów





Oba rozmiary obrazów zachowywały się podobnie, większy z dłuższym czasem wykonania miał bardziej widoczne cechy. Stopień skalowania wpływa na czas, ponieważ shader uruchamiany jest na

każdy piksel z wyjściowego obrazu.
Na wykresie większego obrazu dla skalowania 2 widzimy 3 znaczące wzrosty. Liczba wątków przed wzrostami wynosiła 11 (121 wątki na blok), 16 (256 wątki na blok), 22 (484 wątki na blok). Oznacza to że gdy liczba wątków w bloku jest delikatnie powyżej potęgi dwójki (odpowiednio 128, 256 512) to wydajność spada.