

Dodawanie wektorów

Metody programowania równoległego

Oscar Teeninga

1. Parametry maszyny

Nie miałem dostępu lokalnego do maszyny wyposażonej w kartę graficzną zgodną z CUDA. Dlatego obliczenia wykonywałem zdalnie na PLGRID na Prometheusie.

Tesla K40d	
Liczba multiprocessorów	2880
Taktowanie multiprocessorów	915 MHz
Taktowanie pamięci	3004 MHz
Szerokość szyny danych	384 bit
Wielkość pamięci RAM	11411 MB
Rodzaje pamięci	GDDR5
Maksymalna ilość wątków	1024
Maksymalna ilość bloków	2048

2. Pomiary czasu

W przypadku GPU pomiary czasu były wykonane zgodnie z zaleceniami z UPEL. W przypadku CPU skorzystałem z biblioteki chrono Maksymalna dostępna pamięć do obliczeń to 6GB, a zatem 2^30B, a zatem dla bloku o rozmiarze 2^10 maksymalna liczba gridów to 2^20. Testy przeprowadziłem dla:

- Liczby gridów: 64 128 256 512 1024 2048 4096 8192 16384 32768 65536 131072 262144 teoretyczny zakres: 1~2^31
- Liczby bloków: 64 128 256 512 1024 teoretyczny zakres: 1-1024

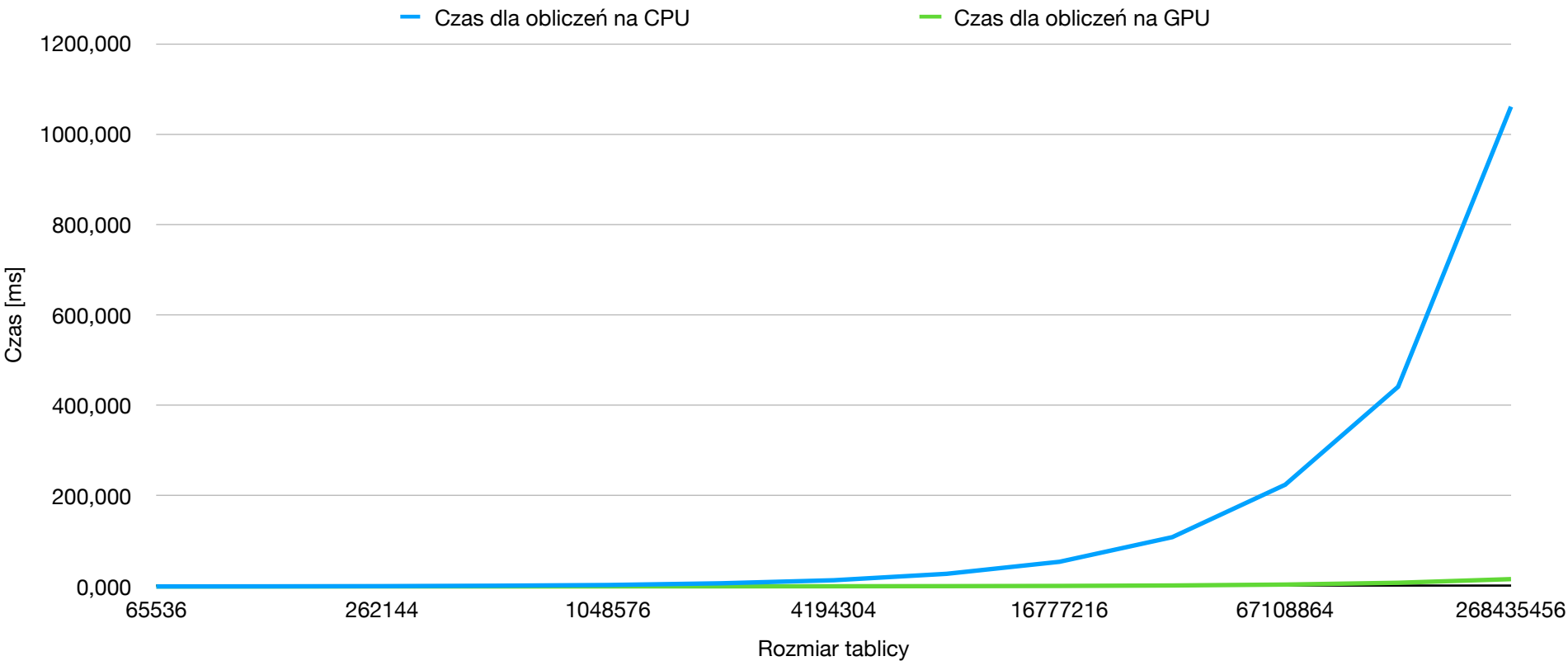
Teoretycznie mógłbym zwiększyć liczbę GRID do 2^20, natomiast na PLGRID nałożony jest jakiś ogranicznik, który powoduje, że zarówno testy dla CPU jak i GPU zostają *Killed* (zapewne zabierają zbyt dużo zasobów).

Czas dodawania wektorów CPU

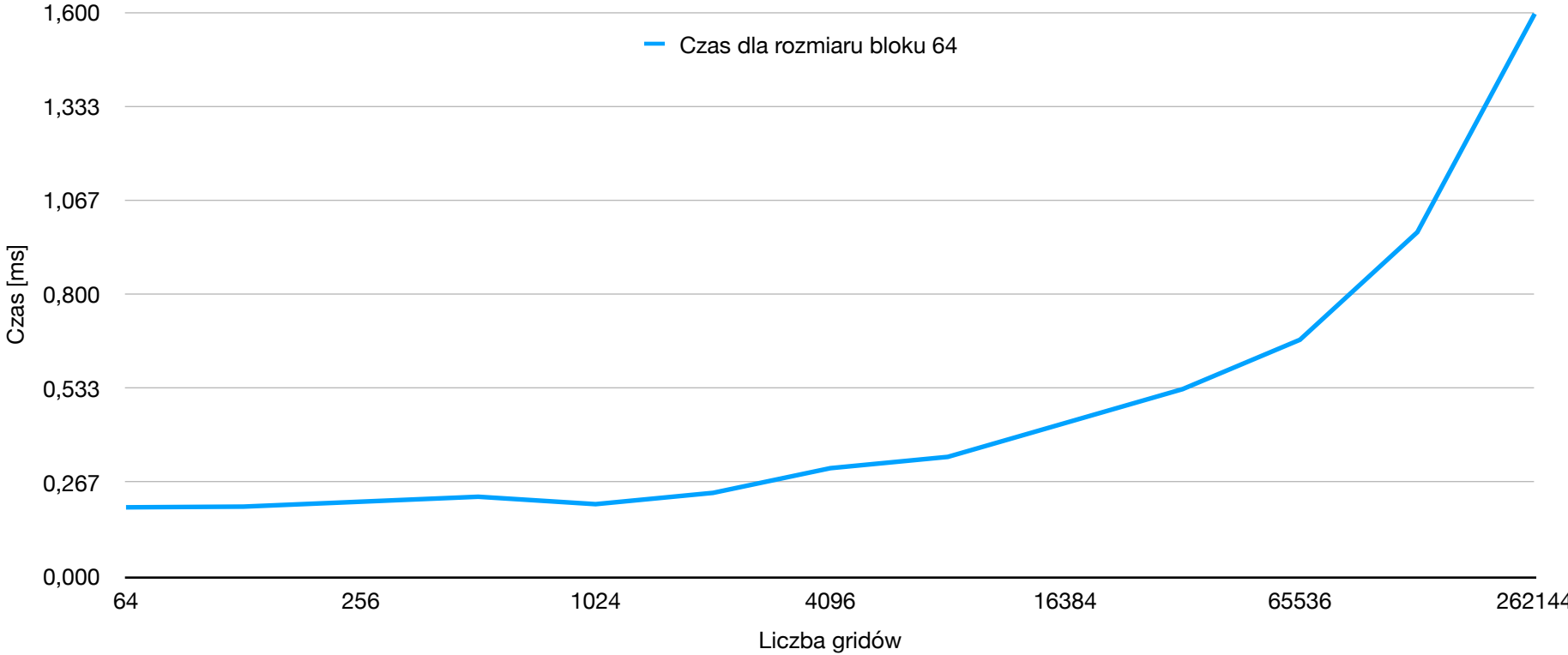
Procesor obliczeniowy	Rozmiar problemu	Czas [ms]
cpu	65536	0,220
cpu	131072	0,420
cpu	262144	0,838
cpu	524288	1,822
cpu	1048576	3,396
cpu	2097152	7,169
cpu	4194304	13,929
cpu	8388608	28,126
cpu	16777216	54,685
cpu	33554432	109,144
cpu	67108864	224,951
cpu	134217728	441,313
cpu	268435456	1059,966

Czas dodawania wektorów GPU

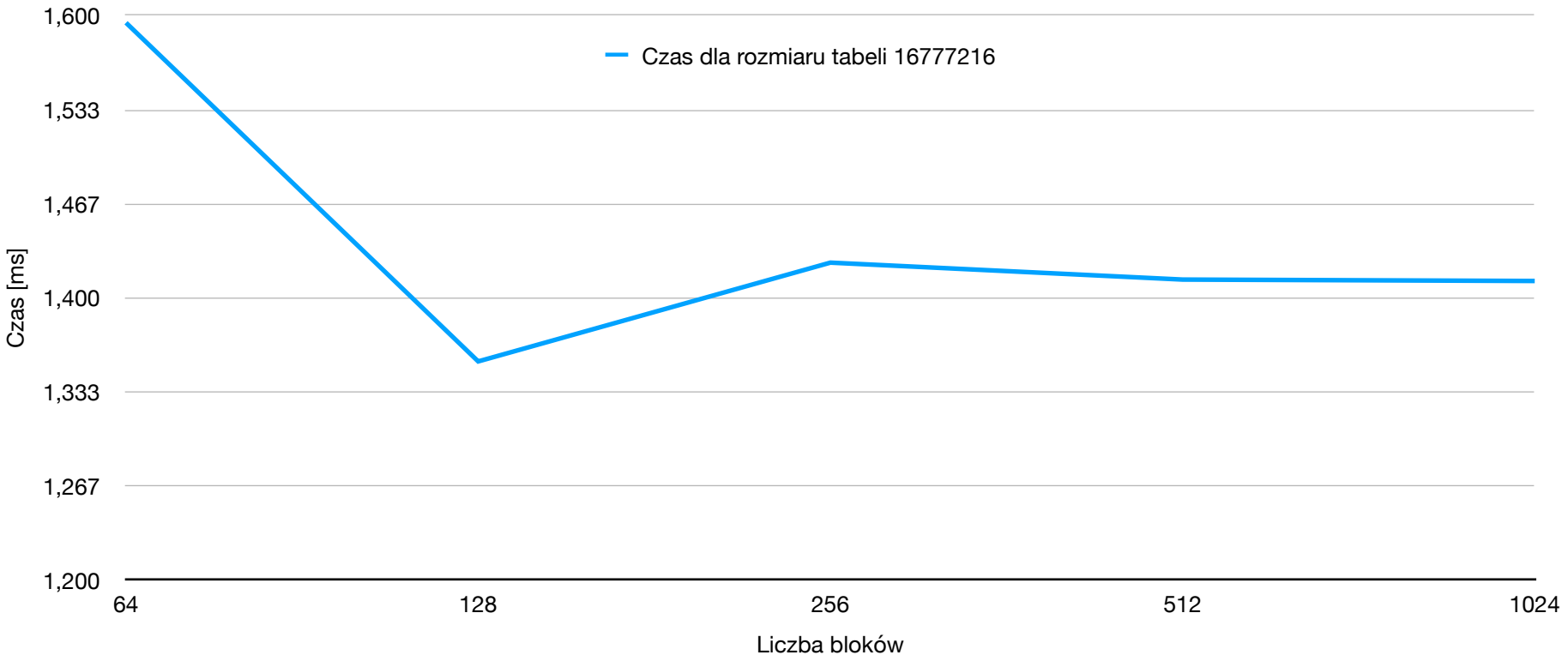
Procesor obliczeniowy	Rozmiar problemu	Czas [ms]
gpu	65536	0,217
gpu	131072	0,227
gpu	262144	0,279
gpu	524288	0,318
gpu	1048576	0,391
gpu	2097152	0,476
gpu	4194304	0,624
gpu	8388608	0,87
gpu	16777216	1,355
gpu	33554432	2,36
gpu	67108864	4,362
gpu	134217728	8,342
gpu	268435456	16,402



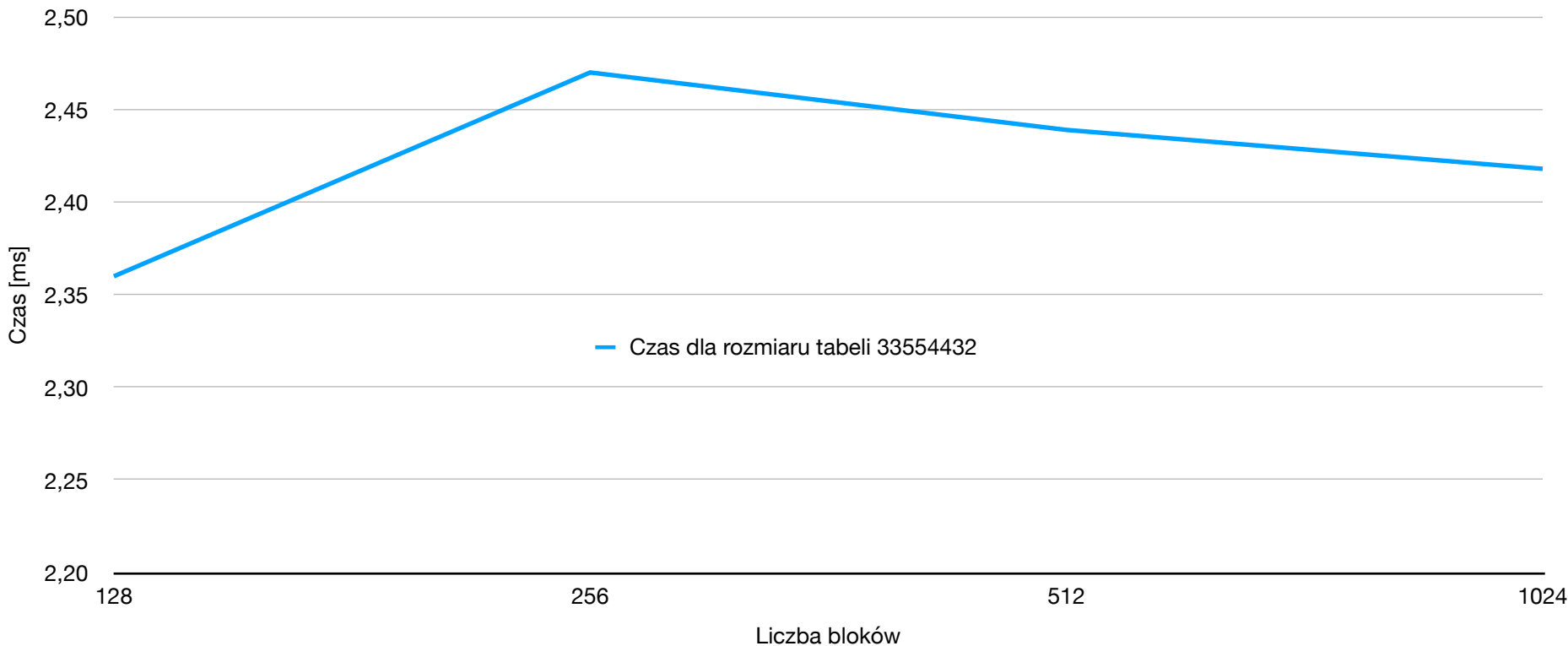
Zależność czasu od liczby gridów dla stałej liczby bloków



Zależność czasu od stosunku gridów do bloków i rozmiarze tabeli 16777216



Zależność czasu od stosunku gridów do bloków i rozmiarze tabeli 33554432



3. Wnioski

Widzimy, że obliczenia wykonywane na GPU są nawet ~60 (!!!)rotnie szybsze niż na CPU. Sytuacja wygląda nieco gorzej jeżeli obliczeniom poddamy również operacja cudaMemcpy oraz cudaMemcpy. Sytuacja ta nie jest zaskakująca, ponieważ procesor graficzny dysponuje znacznie większą liczą rdzeni mogących w tym przypadku bezproblemowo realizować operację dodawania wektorów równoległe. Zauważmy, że dla małych wektorów wyniki dla GPU i CPU są porównywalne, natomiast wraz ze wzrostem rozmiaru tablicy - różnica powiększa się na korzyść GPU.

Jeżeli chodzi natomiast o proporcję liczby bloków per grid, lub liczby gridów, najlepsze efekty osiągnąłem dla 128 bloków per grid oraz N/128 gridów.

Wyniki dla GPU

Procesor obliczeniowy	Rozmiar problemu	Czas [ms]	Liczba gridów	Liczba bloków
gpu	4096	0,196	64	64
gpu	8192	0,198	128	64
gpu	16384	0,212	256	64
gpu	32768	0,226	512	64
gpu	65536	0,205	1024	64
gpu	131072	0,237	2048	64
gpu	262144	0,307	4096	64
gpu	524288	0,339	8192	64
gpu	1048576	0,435	16384	64
gpu	2097152	0,531	32768	64
gpu	4194304	0,671	65536	64
gpu	8388608	0,976	131072	64
gpu	16777216	1,595	262144	64
gpu	8192	0,197	64	128
gpu	16384	0,202	128	128
gpu	32768	0,201	256	128
gpu	65536	0,218	512	128
gpu	131072	0,242	1024	128
gpu	262144	0,306	2048	128
gpu	524288	0,333	4096	128
gpu	1048576	0,404	8192	128
gpu	2097152	0,497	16384	128
gpu	4194304	0,627	32768	128
gpu	8388608	0,889	65536	128
gpu	16777216	1,412	131072	128
gpu	33554432	2,418	262144	128
gpu	16384	0,22	64	256
gpu	32768	0,207	128	256
gpu	65536	0,216	256	256
gpu	131072	0,232	512	256
gpu	262144	0,3	1024	256
gpu	524288	0,332	2048	256
gpu	1048576	0,415	4096	256
gpu	2097152	0,502	8192	256
gpu	4194304	0,626	16384	256
gpu	8388608	0,904	32768	256
gpu	16777216	1,413	65536	256
gpu	33554432	2,439	131072	256
gpu	67108864	4,534	262144	256
gpu	32768	0,223	64	512
gpu	65536	0,202	128	512
gpu	131072	0,236	256	512
gpu	262144	0,302	512	512
gpu	524288	0,338	1024	512
gpu	1048576	0,418	2048	512
gpu	2097152	0,496	4096	512
gpu	4194304	0,636	8192	512
gpu	8388608	0,916	16384	512
gpu	16777216	1,425	32768	512
gpu	33554432	2,47	65536	512
gpu	67108864	4,554	131072	512
gpu	134217728	8,756	262144	512
gpu	65536	0,217	64	1024
gpu	131072	0,227	128	1024
gpu	262144	0,279	256	1024
gpu	524288	0,318	512	1024
gpu	1048576	0,391	1024	1024
gpu	2097152	0,476	2048	1024
gpu	4194304	0,624	4096	1024
gpu	8388608	0,87	8192	1024
gpu	16777216	1,355	16384	1024
gpu	33554432	2,36	32768	1024
gpu	67108864	4,362	65536	1024
gpu	134217728	8,342	131072	1024
gpu	268435456	16,402	262144	1024