Faculté des sciences, **Master TI**

Professeur :
M. LEFER

Rapport OpenCL

CAMI Nicolas

Pau, le 29 mars 2015

Dans ce rapport, nous allons expliquer notre approche pour mener à bien la multiplication de deux matrices carrées :

$$C = A \times B$$

1 Approche CPU

Un méthode simple a été utilisée, avec trois boucles imbriquées. Pour calculer chaque valeur de la matrice C, on parcours une ligne de la matrice A et une colonne de la matrice B.

La complexité est en $O(n^3)$, avec n la largeur des matrices.

2 Approche GPU

Le calcul GPU se fait en fonction de la taille des matrices.

— Si la taille est inférieure ou égale à 1024x1024, on suppose que les matrices peuvent être transférées sur le GPU dans leur totalité. Et donc le calcul peut se faire avec un seul appel au GPU. Dans ce cas, chaque unité de calcul du GPU s'occupe de calculer une case de la matrice C.

La complexité est en $O(n)$, où n est la largeur des matrices.

— Si la taille est supérieure à 1024x1024, on utilise un algorithme de multiplication par blocs. Chaque bloc étant de taille 1024x1024. Il y aura donc plusieurs appels au GPU, et chaque appel au GPU nous ramène au cas précédent (multiplication de matrices de taille 1024x1024).

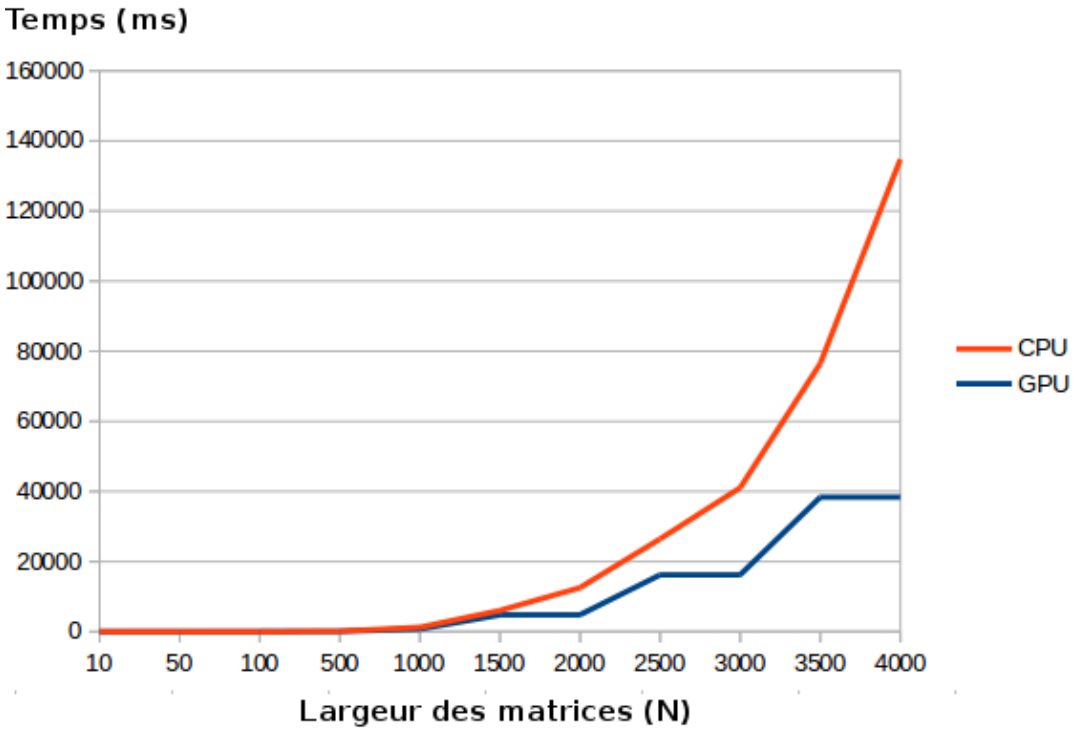
Dans ce cas, la complexité dépendra du nombre de blocs. Il y a m^3.1024 tours de boucles (avec m le nombre de blocs), ce qui fait une complexité en $O(m^3)$.

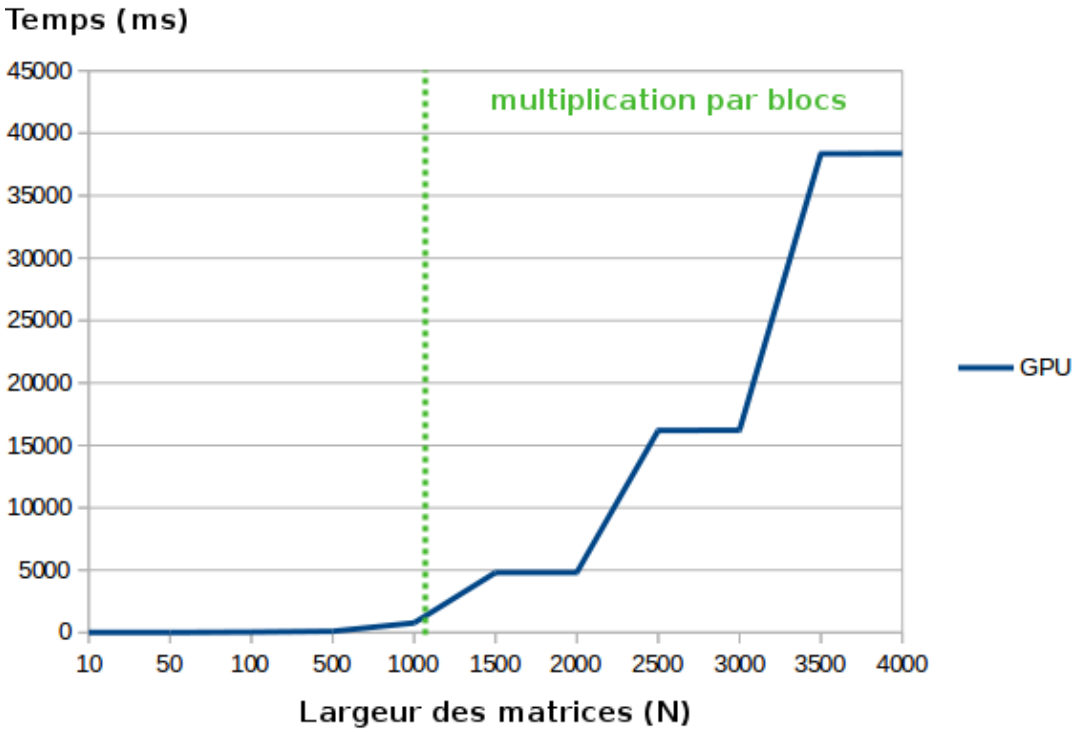
Le nombre de blocs m est défini par : $m = \frac{n}{1024}$ (n est la largeur des matrices).

Dans les deux algorithmes ci-dessus, le même kernel est utilisé sur le GPU.

3 Mesure de performances

Taille des matrices	Temps CPU	Temps GPU
10x10	0.002 ms	1.709 ms
50x50	0.153 ms	1.914 ms
100x100	0.853 ms	45.051 ms
500x500	126.695 ms	98.958 ms
1000x1000	1236.04 ms	768.73 ms
1500x1500	6051.35 ms	4800.53 ms
2000x2000	12539.1 ms	4802.91 ms
2500x2500	26444.7 ms	16192.3 ms
3000x3000	41042.9 ms	16208.8 ms
3500x3500	76447.1 ms	38354.7 ms
4000x4000	134713 ms	38377.8 ms





Sur CPU, le temps d'exécution en fonction de la taille des matrices est exponentiel.

Sur GPU, il est linéaire jusqu'à 1024 (taille d'un bloc de matrice), puis il augmente par palier. Les paliers étant liés au nombre de blocs utilisés pour faire la multiplication de matrices. Plus il y a de blocs, plus il y a de multiplications à faire. Ainsi, la multiplication de matrices de taille 2050x2050 aura le même temps d'exécution que celle de matrices de taille 3000x3000 car dans les deux cas 9 blocs sont utilisés.

Le calcul sur GPU devient vite plus intéressant que sur CPU (ici, dès que la taille des matrices est de 500x500).