TD1 - Open MP

1.1 Vecteur-vecteur

Voici les résultats des temps d'exécution du programme de calcul sur des vecteurs optimisé par OpenMP.

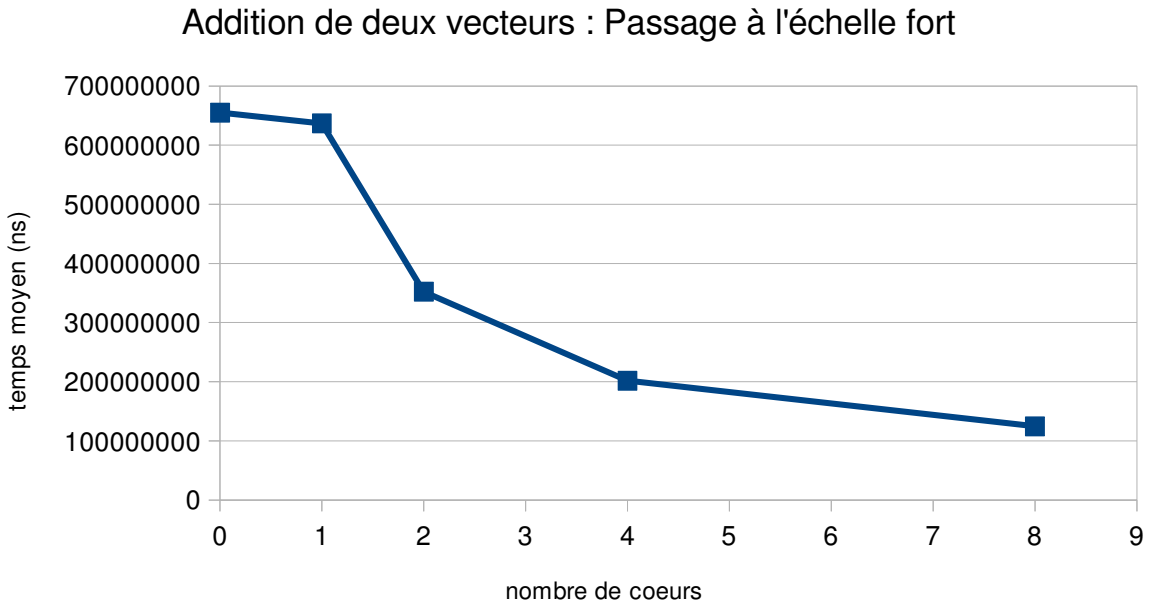
Les valeurs données sont les moyennes des temps sur 10 essais.

Dans les données de passage à l'échelle fort, la valeur avec 0 cœur correspond en réalité à une compilation sans l'option -fopenmp et sans optimisation (-Ox), donc non parallélisée et s'exécutant sur un seul cœur.

Addition de deux vecteurs :

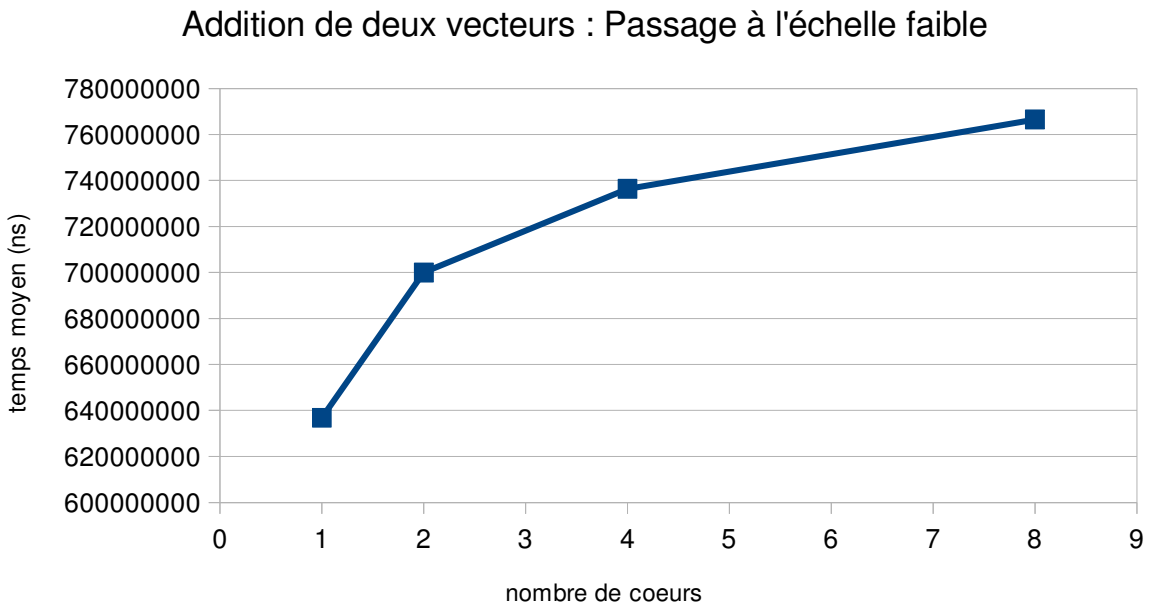
Passage à l'échelle fort : 100 000 000 termes

nombre de coeurs	0	1	2	4	8
temps moyen (ns)	655058308	636827934,6	352213094,4	201747465	124595296,5



Passage à l'échelle faible

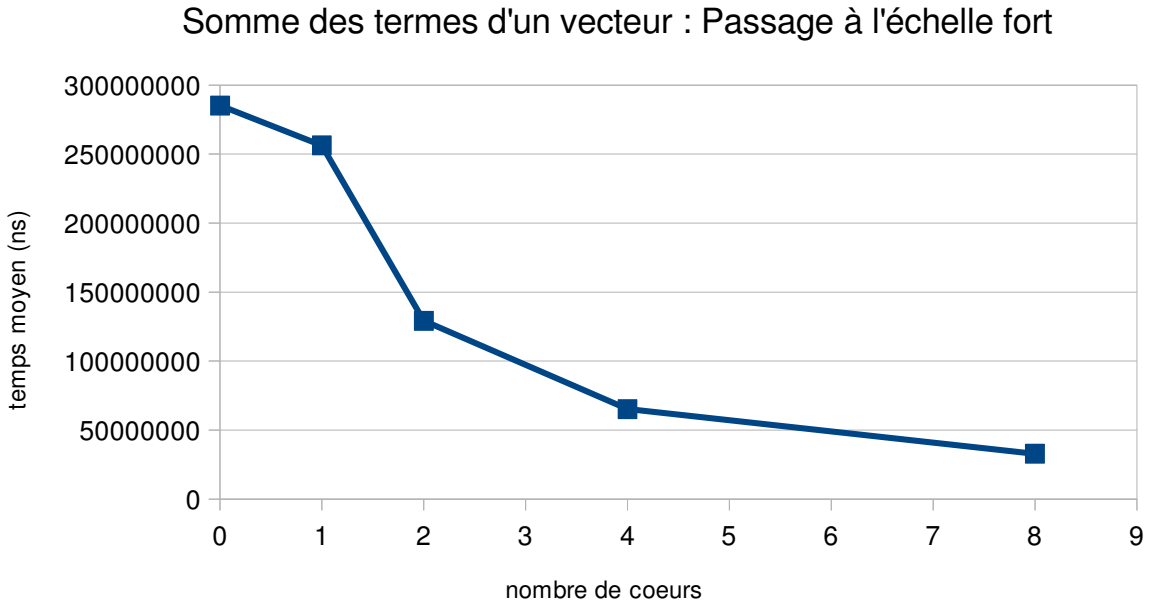
nombre de termes	100000000	200000000	400000000	800000000
nombre de coeurs	1	2	4	8
temps moyen (ns)	636827935	700005867,7	736360163,5	766518690,5



Somme des termes d'un vecteur :

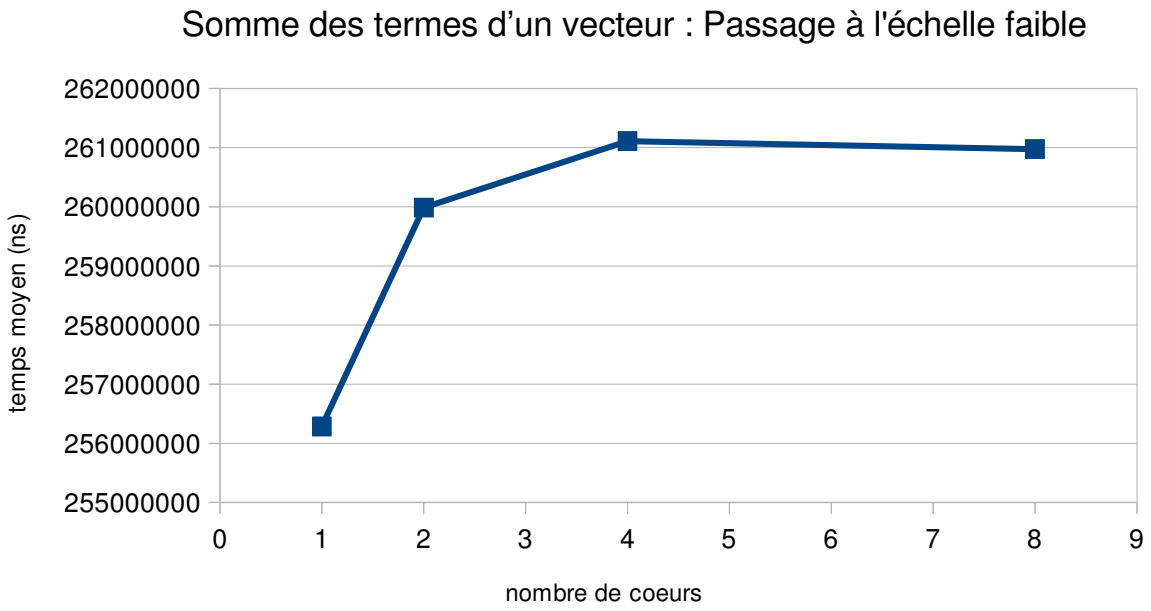
Passage à l'échelle fort : 100 000 000 termes

nombre de coeurs	0	1	2	4	8
temps moyen (ns)	285235055	256286726,4	129158747,7	65162311,5	32824434,8



Passage à l'échelle faible

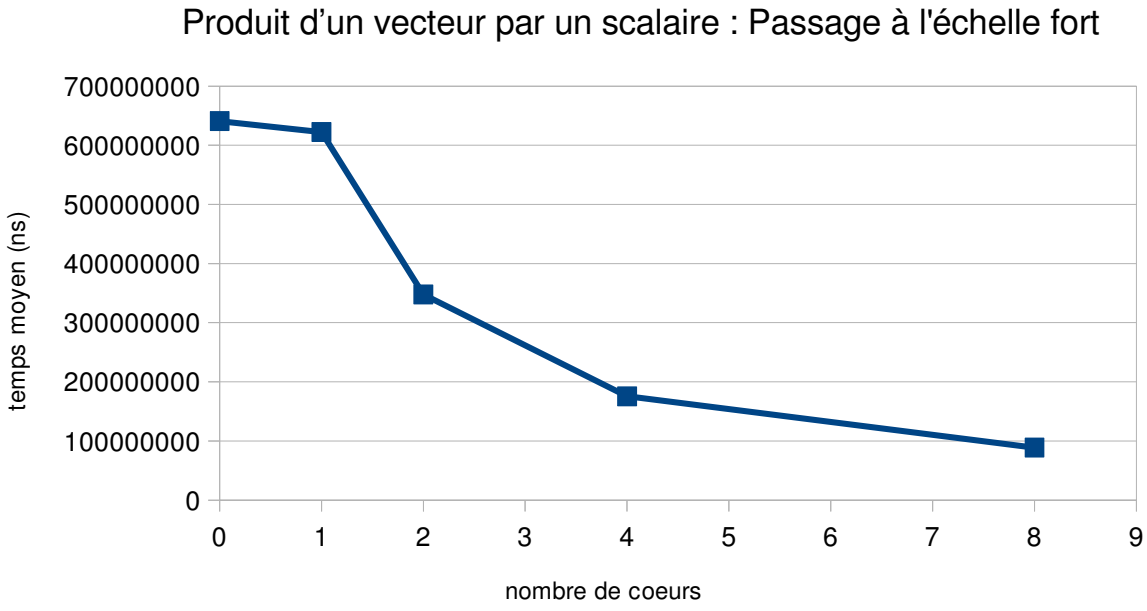
nombre de termes	100000000	200000000	400000000	800000000
nombre de coeurs	1	2	4	8
temps moyen (ns)	256286726	259985866,6	261111140,3	260976126,3



Produit d'un vecteur par un scalaire :

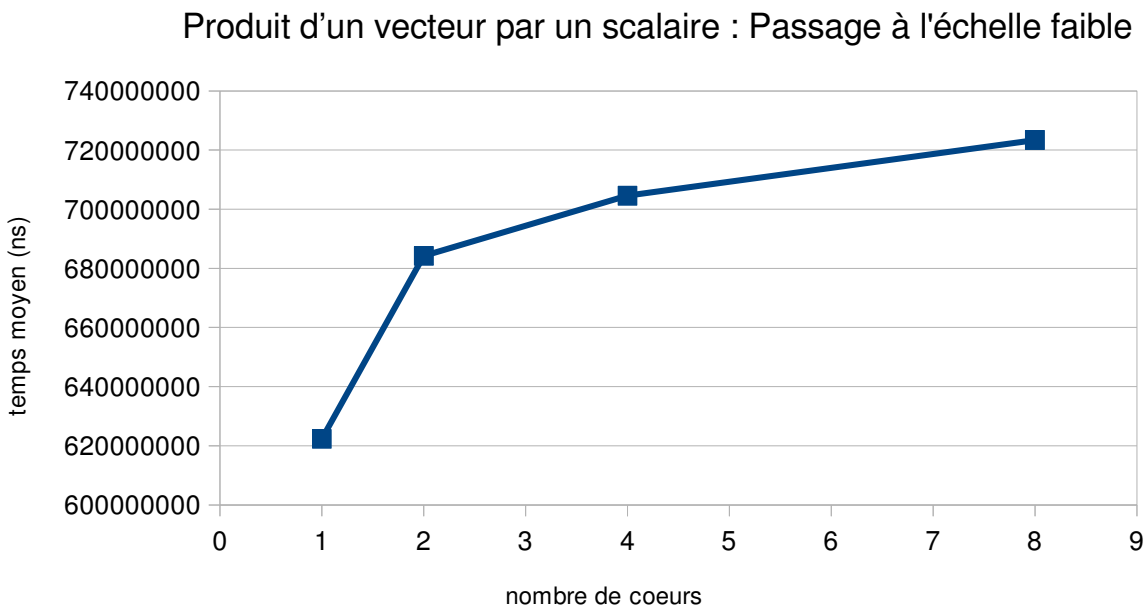
Passage à l'échelle fort : 100 000 000 termes

nombre de coeurs	0	1	2	4	8
temps moyen (ns)	640870684	622340529,2	347700285,3	175592209,4	88787369,8



Passage à l'échelle faible

nombre de termes	100000000	200000000	400000000	800000000
nombre de coeurs	1	2	4	8
temps moyen (ns)	622340529	684203538,1	704566602,1	723410431,1



Globalement, on constate que la compilation avec OpenMP permet une amélioration des performances même lorsqu'on utilise qu'un seul cœur par rapport à une compilation normale non parallélisée. Cela peut s'expliquer par le fait que la compilation OpenMP intègre des optimisation de code à la compilation.

Dans le cas des passage à l'échelle forts, on constate une amélioration des performances qui suit presque le nombre de cœurs utilisé, la vitesse de calcul double presque avec le nombre de cœurs. Le fait est qu'avec l'augmentation du nombre de cœurs, il y a un écart de plus en plus important au résultat attendu si le passage l'échelle fort était parfait (division par 2 du temps de calcul). Il est dû au temps de calcul nécessaire à la création et la destruction des threads OpenMP, augmentant avec le nombre de cœurs.

Enfin avec le passage à l'échelle faible, On peut noter certes une augmentation du temps de calcul par rapport à la taille de la donnée et au nombre de cœurs, dû au temps de synchronisation, de création et de destruction des threads OpenMP, mais que cette augmentation tend à diminuer avec le passage à l'échelle, ce qui sous-tend que OpenMP est particulièrement adapté au passage à l'échelle faible.