

Travaux Pratique 4

Contexte : Les données indiquées dans ce rapport ont été obtenues avec un Processeur 6 cœurs et 12 processeurs logiques.

Exercice 1 :

Size	Tseq	Tpar	Accélération	Efficacité
1024	3	211	0,014218009	0,001185
10240	35	187	0,187165775	0,015597
102400	354	184	1,923913043	0,160326
1024000	3539	957	3,698014629	0,308168
10240000	35791	9725	3,680308483	0,306692
102400000	358000	95428	3,75151947	0,312627

Comme d'habitude, on observe une accélération très faible pour des tailles faibles de vector. C'est dû au temps nécessaire pour instancier des threads qui est assez coûteux. On remarque que l'accélération (et l'efficacité) sont stables avec un grand nombre de données. L'accélération pour n éléments quand n est supérieur à 10^6 est environ 3,7. Ce qui limite l'algorithme pour approcher une accélération proche de p est le bus de la carte mère. Pour résoudre, ce problème, il faudrait que l'opérateur utilisé soit plus complexe et donc plus lent à exécuter. Cela pourrait ainsi occuper plus de temps processeur et laissé plus de temps au transfert des données par la carte mère.

Calcul de la complexité :

On a n valeurs, découpées en p blocs, avec $blockSize = \frac{n+p-1}{p}$

A l'étape 1 : chaque processeur effectue $blockSize-1$ appels de l'opérateur.

A l'étape 2 : 1 seul thread effectue p appels de l'opérateur.

A l'étape 3 : chaque processeur (sauf 1) effectue $blockSize$ appels de l'opérateur.

Par conséquent, on a une complexité en $O(blockSize-1 + p + Blocksize)$.

Comme p est négligeable devant $blockSize$, alors on peut simplifier la complexité par $O(blockSize)$, soit $O\left(\frac{n+p-1}{p}\right)$.

Exercice 2 :

Size	Tseq	Tpar	Accélération	Efficacité
1024	3	204	0,014705882	0,001225
10240	35	199	0,175879397	0,014657
102400	354	196	1,806122449	0,15051
1024000	3570	942	3,789808917	0,315817
10240000	35834	10566	3,391444255	0,28262
102400000	359000	98400	3,648373984	0,304031

On observe exactement la même chose que pour l'exercice 1, ce qui est parfaitement logique car l'algorithme est très proche.

Calcul de la complexité :

On a n valeurs, découpées en p blocs, avec $blockSize = \frac{n+p-1}{p}$

A l'étape 1 : chaque processeur effectue $blockSize-1$ appels de l'opérateur.

A l'étape 2 : 1 seul thread effectue $1+2p$ appels de l'opérateur.

A l'étape 3 : chaque processeur (sauf 1) effectue $blockSize$ appels de l'opérateur.

Par conséquent, on a une complexité en $O(blockSize-1 + 1 + 2p + Blocksize)$.

Comme p est négligeable devant $blockSize$, alors on peut simplifier la complexité par $O(blockSize)$, soit $O\left(\frac{n+p-1}{p}\right)$.