

PROJET DE PROGRAMMATION EN OPENMP
IATIC4-ISTY

(2018-2019)

TRI PARALLELE D'UN TABLEAU

Nahid Emad

On souhaite trier une très grande base de données par un *algorithme parallèle*. On suppose que cette base de données est constituée d'une liste de nombres réels stockés dans N tableaux (blocs) de taille fixe K . Il s'agit donc en total de $N*K$ nombres réels. Pour cela, on se base sur trois fonctions suivantes.

- La fonction *tri-merge* qui accepte en entrée deux blocs triés ***Bin1*** et ***Bin2*** de même taille, crée deux nouveaux blocs ***Bout1***, ***Bout2*** de même taille. Les éléments du bloc ***Bout1*** seront tous plus petits que ceux du bloc ***Bout2***. En sortie de la fonction *tri-merge*, ces deux blocs doivent être triés. Cette fonction pourrait appeler la fonction *tri* suivante.
- La fonction *tri* qui accepte en entrée un bloc ***B*** de taille K et rend en sortie le même bloc ***B*** mais avec les éléments triés.
- La fonction *generator* qui crée un tableau (bloc) de K nombre réels aléatoires.

Algorithme de tri parallèle de $N*K$ nombres réels est présenté ci-dessous :

Entrées :

- N : nombre de tableaux (blocs)
- K : taille des blocs
- ***B1, B2, ..., BN*** : un ensemble de N tableaux (blocs) *non-triés* de taille K

Sorties :

- ***B1, B2, ..., BN*** : un ensemble de N tableaux (blocs) *triés* de taille K

Pour $i=1$ à N faire *en parallèle*

- *tri* (Entrée : ***Bi***, Sortie : ***Bi***)

Fin pour i

Pour $j=1$ à $N-1$ faire

- $k=1+(j\%2)$ // traitement des blocs deux à deux
- Pour $i=0$ à $N/2-1$ faire *en parallèle*
 - $b1=1+(k+2*i)\%N$
 - $b2=1+(K+2*i+1)\%N$
 - $min=min(b1, b2)$
 - $max=max(b1, b2)$
 - *tri_merge* (Entrées : B_{min}, B_{max} , Sorties : B_{min}, B_{max})
- Fin pour i

Fin pour j

Le projet consiste à implémenter en langage C et des directives de compilation openMP, l'algorithme de tri parallèle ci-dessus. Pour cela, il faut commencer par écrire les trois fonctions *tri*, *tri_merge* et *generator*. Votre programme doit être testé pour un nombre total de données ($N \times K$) allant de **10** à **1000000**. Pour chacun des cas, le temps d'exécution doit être mesuré et les courbes de performances doivent être dessinées selon les critères suivants :

- Temps d'exécution en fonction de taille total de données ($N \times K$)
- Pour $N \times K$ fixé, la variation du temps d'exécution en fonction de la variation de K (ou de N)
- Pour $N \times K$ fixé, la variation du temps d'exécution en fonction de nombre de Threads
- Etc.

Pour chaque cas ci-dessus, précisez la charge de chacun des Threads et chacun des cœurs de la machine.

Organisation de travail et rendu du projet

- Le projet doit être fait **en binôme**.
- Un rapport de plus de 6 pages (moins de 10 pages) en format word ou pdf doit être rédigé contenant un résumé de votre travail, les cas tests, les courbes de performances, des charges des threads et des cœurs, votre conclusion ainsi que toutes informations complémentaires que vous jugerez utiles à la compréhension de votre travail.
- Les codes source ainsi que les données des cas test (représentant les courbes de performances présentées dans le rapport) devront être fournis avec le rapport. Attention de ne pas m'envoyer des fichiers binaires (au quel cas une sanction de 3 points vous sera appliquée).
- Les fichiers doivent respecter le format suivant : nom1_prénom1_nom2-prénom2_codeSourceTRI.txt et nom1_prénom1_nom2-prénom2_rapportTRI.pdf (ou word).
- La date limite de l'envoi du projet (rapport et le code source) **est le dimanche 11 novembre 2018 à minuit au plus tard**.