

Série de Travaux Dirigés : Vérification Grille Sudoku - programmation hybride - RMA

Une grille de Sudoku généralisée est une matrice S de taille $k^2 \times k^2$ qui se décompose en $k \times k$ blocs de taille $k \times k$ où chaque bloc contient tous les nombres de 1 à k^2 . On souhaite paralléliser la vérification qu'une grille est bien une grille Sudoku correcte soit que chaque bloc, ligne, colonne contiennent exactement une fois chaque valeur de 1 à k^2 . De plus, on souhaite connaître le résultat le plus vite possible surtout si la grille n'est pas vérifiée.

La distribution de la grille est fixée. Le processeur *root* connaissant la grille complète la décompose en bandes horizontales de taille $k \times k^2$. Et vous pouvez également supposer qu'on dispose de $nprocs$ processeurs tel que $nprocs = k$. Ainsi la distribution de la grille consiste à affecter chaque bande horizontale à un processeur.

Les contraintes à respecter pour votre implémentation sont les suivantes

1. un processeur *root* génère et distribue correctement la grille
2. pour travailler le plus vite possible chaque processeur pourra utiliser **plusieurs threads** pour effectuer la vérification (sur Celene le dossier MPIThreads vous rappellera comment faire).
3. pour la vérification des colonnes vous ne devez pas réunir une colonne sur un processeur mais répartir la vérification. A vous de voir si des communications RMA pourraient faciliter ces dernières vérifications ?
4. vous utiliserez **des communications RMA** pour arrêter la vérification aussi vite que possible dès qu'un processeur trouve qu'une ligne ou un bloc ou une colonne ne vérifie pas le critère.