

Exercice 1. Topologie cartésienne et réduction

L'objectif de cet exercice est de manipuler différents communicateurs pour le calcul d'une somme de scalaires répartis sur des processeurs. Le programme s'exécutera sur $nprocs$ processeurs tel que $nprocs = p \times q$. Les étapes sont les suivantes

1. Créez une topologie cartésienne pour représenter vos processeurs comme une grille de $p \times q$ processeurs.
2. si les coordonnées d'un processeur dans cette grille sont notées (i, j) et que pid désigne son identifiant dans `MPI_COMM_WORLD` initialisez un entier a à la valeur $i \times pid + j$ sur chaque processeur.
3. **En n'utilisant que des réductions** calculez la somme cumulée de tous les entiers a de façon à ce que le processeur `root` soit le seul à avoir le résultat final.

Exercice 2. Produit de matrices par l'algorithme de Cannon

1. Présentation de l'algorithme

L'objectif de cet exercice est d'implémenter l'algorithme de Cannon qui réalise le produit de deux matrices $C = A \times B$ en supposant qu'on travaille à partir d'une grille 2D régulière de processeurs $p \times p$. Les deux matrices sont de taille n^2 et l'algorithme de Cannon est basé sur un découpage des matrices A et B en blocs A_{ij} (respectivement B_{ij}) de taille $\frac{n}{p} \times \frac{n}{p}$ (on supposera que n est divisible par p). Les deux étapes de l'algorithme sont les suivantes

(a) Etape 1 : initialisation

Les matrices sont réparties sur les processeurs tel que le bloc A_{ij} (respectivement B_{ij}) soit sur le processeur identifié par le couple (i, j) sur la grille de processeurs. La première étape consiste à redistribuer la matrice de façon à effectuer un décalage circulaire des blocs A_{ij} de i positions à gauche et des blocs B_{ij} de j positions vers le haut.

Par exemple on doit obtenir cette redistribution pour les matrices A et B découpées en 4×4 blocs :

$$\begin{pmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{pmatrix} \longrightarrow \begin{pmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{11} & A_{12} & A_{13} & A_{10} \\ A_{22} & A_{23} & A_{20} & A_{21} \\ A_{33} & A_{30} & A_{31} & A_{32} \end{pmatrix}$$

$$\begin{pmatrix} B_{00} & B_{01} & B_{02} & B_{03} \\ B_{10} & B_{11} & B_{12} & B_{13} \\ B_{20} & B_{21} & B_{22} & B_{23} \\ B_{30} & B_{31} & B_{32} & B_{33} \end{pmatrix} \longrightarrow \begin{pmatrix} B_{00} & B_{11} & B_{22} & B_{33} \\ B_{10} & B_{21} & B_{32} & B_{03} \\ B_{20} & B_{31} & B_{02} & B_{13} \\ B_{30} & B_{01} & B_{12} & B_{23} \end{pmatrix}$$

(b) Etape 2 : répéter p fois

Chaque processeur identifié par le couple (i, j) sur la grille 2D effectue le produit des deux blocs de A et B qu'il a reçu et les transmet de la manière suivante

- Les blocs de la matrice A sont transmis selon un shift circulaire à gauche dans la grille de processeurs.
- Les blocs de la matrice B sont transmis selon un shift circulaire vers le haut dans la grille de processeurs.

Au final chaque processeur (i, j) contient le bloc $C_{ij} = \sum_{l=0}^{p-1} A_{il} B_{lj}$.

2. Réalisation

On va supposer que n est divisible par p et que le nombre de processeurs est tel que $nprocs = p \times p$. Les étapes pour la mise en œuvre sont les suivantes

- (a) Création d'une topologie 2D des processeurs (`MPI_Dims_create` et `MPI_Cart_create`) pour être capable d'identifier un processeur de rang `pid` par un couple (i, j) dans une grille 2D.
- (b) Initialisation sur chaque processeur de deux matrices locales A_{local} et B_{local} de taille $\frac{n}{p} \times \frac{n}{p}$ qui correspondent aux blocs A_{ij} et B_{ij} des matrices A et B .
- (c) Redistribution des blocs selon l'étape 1 de l'algorithme. Vous pourrez utiliser `MPI_Cart_shift` pour déterminer le rang des processeurs source et destination d'un bloc.
- (d) Mise en place de la boucle à partir d'une fonction qui calcule $C_{local} = C_{local} + A_{local} \times B_{local}$