TP1 Calcul Haute Performance : MPI

Exercice 1 : **Ping-Pong**

On considère un nombre pair de processus. Ecrire un programme MPI qui effectue les tâches suivantes :

- Un processus de rang pair envoye un message contenant son rang au processus impair correspondant (0 envoie à 1, 2 à $3 \ldots$). Il reçoit le message du processus impair associé et l'affiche.
- Un processus de rang impair reçoit le message du processus pair associé. Il envoye un message contenant la valeur reçue plus dix fois son rang.

Exercice 2 : **Anneau**

Prenons un ensemble de processus MPI communiquant en suivant une topologie d'anneau.

1. Si un processus de cet ensemble a le rang k , quel est le rang du processus à qui le processus k va envoyer des messages ?
2. Quel est le rang du processus de qui le processus k va recevoir des messages ?
3. Comment la circulation du jeton doit-elle être initialisée ?
4. Comment la circulation du jeton se termine-t-elle ?
5. Implémentez en MPI une circulation de jeton (un entier) selon un anneau entre des processus. Votre programme doit fonctionner quel que soit le nombre de processus.

Exercice 3 : **Calcul de PI**

Le nombre π peut être défini comme l'intégrale de 0 à 1 de $\frac{4}{1+x^2}$. Une manière simple d'approximer une intégrale est de discrétiser l'ensemble d'étude de la fonction On considére l'approximation suivante avec $s = \frac{1}{n}$:

$$\pi \approx \int_0^1 \frac{4}{1+x^2}\,\mathrm{d}x \approx \sum_{i=0}^{n-1} s \times \frac{f(i \times s) + f((i+1) \times s)}{2}$$

On peut alors répartir le calcul de π en répartissant l'intervalle entre p processus.
Proposez un programme MPI qui parallélise cette approximation de la valeur de π.

Exercice 4 : **Produit matrice vecteur en parallèle**

Le but de cet exercice est de proposer et étudier un code parallèle pour le calcul d'un produit matrice vecteur $y = Ax$, où A est une matrice dense de taille $n \times n$. Nous considérons que la matrice A et le vecteur x sont initialisés par le processus de rang 0 puis la matrice A est distribuée au long de ses lignes (1D) sur p processus et le vecteur x envoyé à tous les processus.
A_i représente un block de la matrice A de taille $n/p \times n$ attribué au processus p_i.

1. Ecrivez un programme séquentiel calculant le produit matrice vecteur.
2. Ecrivez un programme parallèle MPI calculant le produit matrice vecteur en considérant une distribution au long des lignes de la matrice initiale A.
3. Analysez le changement introduit par une distribution au long des colonnes de la matrice A.

Exercice 5 : **Tri Pair Impair**

Soit un tableau d'entiers de taille n dont on souhaite effectuer un tri dans l'ordre croissant. A partir de l'exemple ci-dessous, proposez une implémentation en MPI de ce tri parallèle en supposant que n est proportionnel au nombre p de processus.

L'algorithme proposé est basé sur une distribution du tableau à trier sur les p processus qui lors d'une première étape trie leur partie du tableau de taille n/p

Ensuite en p étapes les processus terminent le tri de la manière suivante :

1. si l'étape est paire :
   (a) les processus de rang pair transmettent au processus de droite, recoivent du processus de droite et réalisent une fusion des deux tableaux en gardant les n/p plus petits éléments.
   (b) les processus de rang impair transmettent au processus de gauche, recoivent du processus de gauche et réalisent une fusion des deux tableaux en gardant les n/p plus grands éléments.
2. si l'étape est impaire :
   (a) les processus de rang pair transmettent au processus de gauche, recoivent du processus de gauche et réalisent une fusion des deux tableaux en gardant les n/p plus grands éléments.
   (b) les processus de rang impair transmettent au processus de droite, recoivent du processus de droite et réalisent une fusion des deux tableaux en gardant les n/p plus petits éléments.

Par exemple :