

## TP3 Programmation Parallèle : MPI

### Exercice 1 : Ping-Pong

Ecrire un programme MPI qui effectue les tâches suivantes:

- Processus de rang pair : envoyer un message contenant le rang du processus courant au processus impair correspondant (0 envoie à 1, 2 à 3 ...). Recevoir le message du processus impair et l'afficher.
- Processus de rang impair : recevoir le message du processus pair associé; Envoyer un message contenant la valeur reçue plus dix fois le rang du processus courant.

### Exercice 2 : Calcul de PI

1. Proposez un programme MPI qui parallélise une approximation de la valeur de PI.

### Exercice 3 : 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  $y = y + Ax$ , où  $A$  est une matrice dense. 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.

1. Distribuer les lignes de  $A$  tel que chaque processus reçoit  $n/p$  lignes de  $A$ . Utiliser `MPI_Scatter`.
2. Distribuer le vecteur  $x$ . Utiliser `MPI_Broadcast`.
3. Effectuer le produit matrice-vecteur local  $y_{local} = A_{local}x$  ( $y_{local}$  est de taille  $n/p$ ).
4. Mettre ensemble le vecteur  $y$  entier dans le processus 0. Utiliser `MPI_Gather`. Quel changement faudrait-il pour que tous les processus possède le résultat  $y$ ? Faire la modification nécessaire dans le code.
5. Analysez le changement introduit par une distribution au long des colonnes de la matrice.

### Exercice 4 : 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 part de 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 numéro pair transmettent au processus de droite, reçoivent du processus de droite et réalisent une fusion des deux tableaux en gardant les  $n/p$  plus petits éléments.
  - (b) les processus de numéro impair transmettent au processus de gauche, reçoivent 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 numéro pair transmettent au processus de gauche, reçoivent du processus de gauche et réalisent une fusion des deux tableaux en gardant les  $n/p$  plus grands éléments.
- (b) les processus de numéro impair transmettent au processus de droite, reçoivent du processus de droite et réalisent une fusion des deux tableaux en gardant les  $n/p$  plus petits éléments.

Par exemple :

