

TP5 - Applications parallèles MPI et communications collectives

1 But du TP

L'objectif de ce TP est de vous initier au développement d'applications parallèles MPI en utilisant les primitives de communication collective.

2 La primitive *MPI_Bcast*

Ecrivez, en utilisant la primitive *MPI_Bcast* un programme dans lequel le processus de rang 0 envoie un message à tous les autres processus du groupe.

3 La primitive *MPI_Scatter*

On considère un tableau d'entiers de taille fixe N . On se propose ici de fragmenter le tableau en sous-tableaux de taille égale et de les envoyer à tous les processus du groupe. Chaque processus, y compris le processus de rang 0, doit recevoir un seul sous-tableau. Si N n'est pas multiple de la taille du groupe (en nombre de processus), il faut fabriquer artificiellement des éléments supplémentaires (en les initialisant à 0).

- Développez un programme (version 1) réalisant le traitement demandé en utilisant les primitives *MPI_Send* et *MPI_Recv*.
- Ecrivez un autre programme (version 2) réalisant le même traitement mais cette fois-ci en utilisant les primitives *MPI_Bcast* et *MPI_Scatter*.
- Mesurez le temps d'exécution des deux programmes et comparez-les. Quelle conclusion tirez-vous ?

4 La primitive *MPI_Reduce*

- Modifiez la version 2 en version 3 telle que le processus 0 calcule la somme de tous les éléments du tableau, en récupérant la somme partielle effectuée par chaque processus sur son sous-tableau. Il est demandé ici d'utiliser la primitive *MPI_Reduce*.
- Comparez les temps d'exécution des trois versions.

5 La primitive *MPI_Gather*

Ecrivez un programme permettant de modifier le tableau d'entiers en ajoutant un nombre donné à chaque élément du tableau. Utilisez les primitives *MPI_Bcast*, *MPI_Scatter* et *MPI_Gather*.