

TP 2 – Systèmes et Traitements Répartis GIS 5

Fonctions classiques de MPI

Exercice 1 : Envoyer le même message à plusieurs processus

`MPI_Bcast` est une fonction *One-to-all* où un processus partage une même donnée à tous les processus (lui y compris). Par exemple, `MPI_Bcast(message, length, MPI_CHAR, 0, MPI_COMM_WORLD)` ; signifie que le processus de rang 0 envoie la variable `message` de type `MPI_CHAR` et de taille `length` à chaque processus inclus dans le groupe de processus `MPI_COMM_WORLD`.

Attention : pour un envoi de chaîne de caractère, il faut compter le caractère `'\0'`, ignoré par la fonction `strlen()`.

Écrire un code utilisant `MPI_Bcast` où un processus envoie aux autres la date d'aujourd'hui, allouée dynamiquement. Les processus la recevant doivent afficher un message indiquant leur rang et la date reçue. On remarque que la taille de la date n'est pas connue (à moins que ?...).

Exercice 2 : Distribuer des messages différents à plusieurs processus

`MPI_Scatter` est une fonction *One-to-all* où un processus diffuse une donnée sur tous les processus (lui y compris). `MPI_Scatter(T, 10, MPI_INT, local_T, 10, MPI_INT, proc, MPI_COMM_WORLD)` ; signifie que le processus `proc` répartit des morceaux du tableau `T` par paquets de 10 éléments sur les tableaux `local_T` de chaque processus inclus dans la groupe de processus.

Écrire un code où un processus crée un tableau aléatoire de taille 1000 et envoie aux autres processus une partie différente de ce tableau avec `MPI_Scatter`. Chacun des processus affichera le morceau de tableau reçu.

Exercice 3 : Rassembler les messages de plusieurs processus

`MPI_Gather` est une fonction *All-to-one* où un processus rassemble les données des autres (lui y compris). `MPI_Gather(partial_T, N, MPI_INT, T, N, MPI_INT, proc, MPI_COMM_WORLD)` ; signifie que `proc` va joindre chaque tableau `partial_T` de `N` éléments qu'il reçoit pour en former plus grand : `T`.

On reprend le code de l'exercice 2. On veut ensuite que chaque processus calcule la somme des éléments de `local_T` et envoie ensuite son résultat au processus de rang 0 dans un nouveau tableau `U`. On calculera ensuite la somme des éléments de `U`.

Exercice 4 : Réduction et opérations directes

`MPI_Reduce` est une fonction *All-to-one* où un processus fait une opération sur les données qu'il reçoit des autres. Quelques exemples d'opérations possibles avec `MPI_Reduce` : `MPI_SUM` (somme les éléments), `MPI_PROD` (multiplie les éléments), `MPI_MAX` (cherche le maximum des éléments), `MPI_MIN` (cherche le minimum des éléments).

`MPI_Reduce(&terme, &resultat, 1, MPI_INT, MPI_PROD, proc, MPI_COMM_WORLD)` ; signifie que `proc` va effectuer le produit de tous les terme reçus et les stockera dans `resultat`.

Faire l'exercice 3 en utilisant `MPI_Reduce`.