

Un hello World parallèle

En testant le programme HelloWorld, on constate que l’affichage des processus n’est pas à l’ordre des numéros des processus.

Par contre, en testant le programme SkeletonMPIProgramWithFilesOutput, les résultats sont dans les fichiers. C’est plus claire qu’une sortie sur terminal, car on peut voir clairement la sortie de chaque processus.

Envoi bloquant et non bloquant

La première version est sûre, car MPI_Send est bloquant, où la modification de la valeur de myvar après MPI_Send ne change pas l’envoi de la valeur de myvar.

Par contre, la deuxième version est dangereux, car MPI_Isend est non bloquant, où la modification de la valeur de myvar après MPI_Isend peut changer l’envoi de la valeur de myvar.

Pour que la deuxième version soit sûre, il faut mettre MPI_Wait juste après MPI_Isend.

Circulation d’un jeton dans un anneau

A l’exécution, les messages sont affichés dans l’ordre des numéros des processus. C’est parce que l’on utilise MPI_Send et MPI_Recv, qui sont bloquants. Les processus doivent tout d’abord recevoir le jeton du processus précédent et ensuite l’envoyer au processus suivant.

Après modifier le programme, on constate qu’il y a un problème : on ne peut qu’entrer la valeur du jeton dans le premier processus. La raison est que scanf ne marche que dans les programmes séquentiels, mais pas dans les programmes parallèles.

Calcul de pi par lancer de fléchettes

Diffusion d’un entier dans un réseau hypercube

Je pense que l’idée est de diffuser le jeton aux points voisins pour chaque point, mais je ne sais pas comment le faire dans les codes... Pour moi, la difficulté est de trouver une façon pour connaître les points voisins et éviter répétition.