

Langages du parallélisme

TP 4 (séance 5)

Exercice 1: MPI-RMA, synchronisation globale

Ecrire un programme dans lequel deux processus écrivent un entier dans la mémoire dans même processus à l'aide d'une communication one-sided. Vous utiliserez le mode `Fence`, puis `lock_all/unlock_all`

Exercice 2: MPI-RMA, lock/unlock et PSCW

Ecrire un programme dans lequel un processus écrit un entier dans la mémoire d'un autre processus à l'aide d'une communication one-sided. Vous utiliserez le mode `Post-Start-Complete-Wait`, puis `lock/unlock`.

Exercice 3: MPI-RMA

Ecrire un programme dans lequel un processus récupère un entier dans la mémoire dans même processus à l'aide d'une communication one-sided, puis l'incrémente de 1 et l'affiche. Vous utiliserez le mode que vous voulez.

Exercice 4: Echange de données

Ecrire un programme dans lequel un processus envoie un tableau de 3 entiers et un double à un autre processus. Pour cela, vous devrez créer un nouveau type de données.

Exercice 5: Diffusion d'une sous-matrice

Soit une matrice $n \times m$ notée $A(n, m)$ créée par le processus 0. Elle pourra être lue en mémoire ou calculée. On suppose qu'on a 4 processus et que le processus 0 va envoyer une partie de la matrice à chaque processus. Le processus 1 reçoit $A(i, j)$ pour $i = n/2 + 1, \dots, n$, et $j = 1, \dots, m/2$. Le processus 2 reçoit $A(i, j)$ pour $i = 1, \dots, n/2$ et $j = m/2 + 1, \dots, m$ et le processus 3 reçoit $A(i, j)$ pour $i = n/2 + 1, \dots, n$ et $j = m/2, \dots, m$. Utilisez `MPI_Scatterv` pour envoyer les données du processus 0 à tous les autres processus (incluant le processus 0). Par simplicité, vous pourrez choisir $n = m = 8$.

Exercice 6:

Finir les exercices des autres TPs.