ENSEIRB-MATMECA

Parcours CISD

Langages du parallélisme
TP 2 (séance 3)

Exercice 1: Barrière de synchronisation

Ecrire un programme dans lequel les processus effectuent une barrière de synchronisation. Utilisez des boucles ou des fonctions `sleep` pour faire varier les temps d'exécution sur les différents processus.

Exercice 2: Diffusion

Ecrire un programme dans lequel le processus 0 envoie un tableau d'entiers à tous les autres processus.

Exercice 3: Réduction

Ecrire un programme dans lequel tous les processus font la somme d'une partie d'un tableau et l'envoient à un processus qui se chargera de faire la somme finale.

Exercice 4: Rassemblement

Ecrire un programme dans lequel un processus récupère un float de la part de tous les autres processus.

Exercice 5: Somme globale

Ecrire un programme où tous les processus calculent la somme de tous les identifiants des processus et envoient ce résultat sur tous les processus.

Exercice 6: Coût des collectives

Réalisez des petits programmes de mesure afin de vous faire une idée des performances des collectives. Vous pourrez également faire un graphique regroupant les temps moyen des différentes opérations en fonction du nombre de processus. Votre code ressemblera à:

```
1 time_start = timer();
2 for (i=0 ; i < iterations; i++) {
3   collective(msg_size);
4   barrier();
```

```
5 }
6 time_end = timer();
7 time = (time_end - time_start) / iterations;
```

Testez les collectives suivantes: `MPI_Barrier`, `MPI_Bcast`, `MPI_Reduce`, `MPI_Allreduce`, `MPI_Alltoall`, `MPI_Scatter`, `MPI_Gather`.

Exercice 7: Communication dans un anneau

Ecrire un programme MPI où parmi n processus, le processus de rang r reçoit la valeur $1000 + (r - 1)$ du processus de rang $r - 1$, $1 \leq r \leq n - 1$, et où le processus de rang 0 reçoit la valeur $1000 + (n - 1)$ du processus de rang $n - 1$.