ENSTA Paris – AMS301 – Calcul scientifique parallèle – Année 2021-2022

TP 1 : Fonctionnalités MPI de bases

L'objectif de cette séance est d'apprendre à utiliser les fonctionnalités de base de MPI. Les exercices seront réalisés à distance sur les stations de travail de l'école. Les codes de départ sont disponibles sur le site Internet du cours.

Exercice 0. Connexion à distance

Passez en revue le tutoriel *"Utilisation à distance des stations de travail de l'ENSTA"*

Exercice 1. Première utilisation de MPI

On fournit deux versions du programme helloworld (`helloworld_mpi.c` et `helloworld_mpi2.c` en `C`, avec l'extension `.cpp` en `C++`). Ouvrez ces fichiers et vérifiez votre compréhension de chacune des lignes. Ensuite, compilez et exécutez les programmes avec les commandes qui suivent. Comprenez-vous le résultat pour chacun des cas ?

Version `C`

```
mpicc helloworld_mpi.c; ./a.out
mpicc helloworld_mpi.c; mpirun -np 2 ./a.out
mpicc helloworld_mpi.c; mpirun -np 4 ./a.out
mpicc helloworld_mpi.c; mpirun ./a.out
gcc helloworld_mpi.c; ./a.out
```

Version `C++`

```
mpicxx helloworld_mpi.cpp; ./a.out
mpicxx helloworld_mpi.cpp; mpirun -np 2 ./a.out
mpicxx helloworld_mpi.cpp; mpirun -np 4 ./a.out
mpicxx helloworld_mpi.cpp; mpirun ./a.out
g++ helloworld_mpi.cpp; ./a.out
```

Exercice 2. Communications point-à-point bloquantes

On fournit un programme (`exchange_mpi.c` en `C`, avec l'extension `.cpp` en `C++`) où deux tableaux sont échangés par les deux premiers processus MPI en utilisant les fonctions `MPI_Send` et `MPI_Recv`. Ainsi, chacun de ces processus envoie un tableau et en reçoit un autre.

On vous demande de tester ce programme en modifiant l'ordre d'appel des fonctions d'envoi et de réception, pour des tableaux de petite taille (10) et de grande taille (10000).

	Processus 0	Processus 1
Cas 1	`MPI_Send`, puis `MPI_Recv`	`MPI_Recv`, puis `MPI_Send`
Cas 2	`MPI_Recv`, puis `MPI_Send`	`MPI_Send`, puis `MPI_Recv`
Cas 3	`MPI_Recv`, puis `MPI_Send`	`MPI_Recv`, puis `MPI_Send`
Cas 4	`MPI_Send`, puis `MPI_Recv`	`MPI_Send`, puis `MPI_Recv`

Comprenez-vous le résultat pour chacun des cas ?

Exercice 3. Schema de communication en anneau

Implémentez un programme où une valeur entière (nommée *jeton*) est envoyée d'un processus MPI à un autre, de manière cyclique, en utilisant les fonctions `MPI_Send` et `MPI_Recv`. À chaque passage du jeton par un processus MPI, sa valeur est incrémentée, et le processus affiche son rang et la valeur du jeton.

Exemple pour un cas avec 3 processus MPI :

- le processus de rang 0 initialise le jeton à '0' et l'envoie au processus de rang 1 ;
- le processus de rang 1 incrémente le jeton à '1' et l'envoie au processus de rang 2 ;
- le processus de rang 2 incrémente le jeton à '2' et l'envoie au processus de rang 0 ;
- le processus de rang 0 incrémente le jeton à '3' et l'envoie au processus de rang 1 ;
- etc.