

TP M54 : Introduction à MPI.

Laurent Desbat

Octobre 2002

1. Mettre en oeuvre le programme QuiSuisJe.c vu en cours (programme dans lequel chaque processus s'identifie).
2. Mettre en oeuvre un programme de communication point à point sur un anneau dans lequel le processeur i envoie les i premiers entiers naturel et les i réels (float) $1/j, j = 1, \dots, i$ au processus $(i + 1) \bmod(n_{\text{proc}})$ où n_{proc} est le nombre total de processus (le choisir pour l'exécution égal à 5).
3. Ecrire un programme dans lequel le processus root diffuse de manière sélective n_{proc} séries de $N = 1000$ doubles $x_i, i = 1, \dots, N$ sur lesquels il faut calculer la moyenne $\bar{x} = 1/N \sum_{i=1}^N x_i$, la moyenne des écarts à la moyenne à la puissance p : $1/N \sum_{i=1}^N (x_i - \bar{x})^p$, pour $p = 1, \dots, 50$. Le processus root collecte ensuite les résultats puis les affiche.

Ne pas oublier de mettre `#include "mpi.h"` en tête de fichier c!

Syntaxe :

```
int MPI_Send(void* buf, int count, MPI_Datatype datatype, int dest,
int etiquette, MPI_Comm communicateur)
int MPI_Recv(void* buf, int count, MPI_Datatype datatype, int source,
int etiquette, MPI_Comm communicateur, MPI_Status *status)
```

Exemple d'utilisation :

```
MPI_Status status;
int tableau[100], proc, etiquette;
/* reception de 50 entiers du processus proc */
MPI_Recv(tableau, 50, MPI_INT, proc, etiquette, MPI_COMM_WORLD, &status)
```

```

#include<stdio.h>
#include <stdlib.h>
#include<mpi.h>

int main(int argc, char *argv[])
/* Mon premier programme en MPI C
compilation : mpicc -o mpifileC mpifileC.c
execution : dmpirun -np 5 mpifile */
{
    int rank, size; /* numero du processus ; nombre de processus */
    MPI_Status status;
    char SendMsg[]="Hello from processor 0"; /* message à envoyer */
    char RecvMsg[50]; /* buffer de reception */
    int root, tag, rankdest; /* maitre (0), etiquette, destinataire */
    int i;

    /* initialization */
    MPI_Init(&argc,&argv);

    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&size);
    /* Qui Suis-je ? */
    printf(" je suis le processus %d, parmi %d processus \n",rank,size);

    /* envoi de message */
    root = 0; /* envoie un message à tous les processeurs avec MPI_Send */
    if(rank==root)
        for(i=1;i<size;i++){
            tag=i;
            rankdest=i;
            MPI_Send(SendMsg,23,MPI_CHAR,rankdest,tag,MPI_COMM_WORLD);
        }
    /* reception du message */
    if(rank!=root) {
        tag = rank;
        MPI_Recv(RecvMsg,50,MPI_CHAR,root,tag,MPI_COMM_WORLD,&status);
        printf("Je suis le processus %d : message recu : %s \n",rank,RecvMsg);
    }
    MPI_Finalize();
}

```