

Ejercicios MPI

Dejad las soluciones dentro de vuestro *home* en una carpeta llamada ejercicios/nombre-tema. Cada ejercicio lo podéis llamar, por ejemplo, ejX.Y.c. En todos ellos la transmisión se debe realizar utilizando tipos derivados.

1 Definición de tipos derivados

1. Escribe un programa MPI que envíe una fila (seleccionada de forma aleatoria) de una matriz de dimensiones 16x16 (inicializada de forma aleatoria) al resto de procesos.
2. Modifica el programa anterior para que envíe una columna en vez de una fila.
3. Escribe un programa MPI que envíe los elementos superiores (sin incluir los de la diagonal principal) de una matriz de dimensiones arbitrarias (generada e inicializada de forma aleatoria) al resto de procesos.
4. Modifica el programa anterior para que envíe los elementos inferiores (sin incluir los de la diagonal principal).
5. El proceso 0 de un programa MPI tiene que enviar al resto de procesos un *array* de enteros **E** de longitud igual al numero de procesos que lo componen, un entero que representa esa longitud y un *array* de *floats* **F** del mismo tamaño. Cada proceso tiene que realizar la operación $res = \sum E(i) * F(i)$ y el proceso 0 tiene que imprimir la suma de todos los resultados parciales (res). Escribe el código en MPI necesario para realizar el proceso anterior.

2 Comunicadores

1. Desarrollar un programa MPI para multiplicar matrices de forma paralela. La manera en que se distribuya el cómputo entre los procesadores es libre pero hay que intentar que sea eficiente. El programa recibirá como parámetros el tamaño de las matrices (cuadradas) y utilizará reserva dinámica de memoria. El proceso *root* inicializará las matrices con valores aleatorios de tipo *float* y repartirá los datos siguiendo el criterio decidido. El código se ha de documentar y hay que explicar el método seguido por el algoritmo implementado.

3 Entrada/Salida

1. Modificar el programa el desarrollado en punto 2.1 para que las matrices, en vez de ser generadas de forma dinámica, sean leídas desde un fichero. Hay que hacer dos versiones:
 - El proceso *root* lee los datos y los distribuye usando funciones MPI de comunicación estándar. Tras realizar la multiplicación, el proceso *root* debe guardar el resultado en un fichero.
 - Los procesos leen los datos desde el fichero de entrada y guardan el resultado en otro fichero de salida utilizando las funciones específicas de E/S que proporciona MPI.

En ambos casos hay que medir el tiempo necesario para completar todo el proceso e imprimirlo por salida estándar. Además, hay que generar un programa independiente para generar el fichero de entrada que contiene las matrices. Este programa debe permitir generar matrices de tamaño arbitrario.