Práctica

3

Multiprocesadores

Utilización de la librería de paso de mensajes MPI.

En esta práctica se trabajará con las comunicaciones colectivas.

Ejercicio 1. Broadcast y barreras

Toma el programa de los trapezoides y modifícalo levemente para que el proceso raíz tome los datos de entrada y los transmita con un *broadcast* a los esclavos, comprobando antes y después del envío que se ha realizado correctamente.

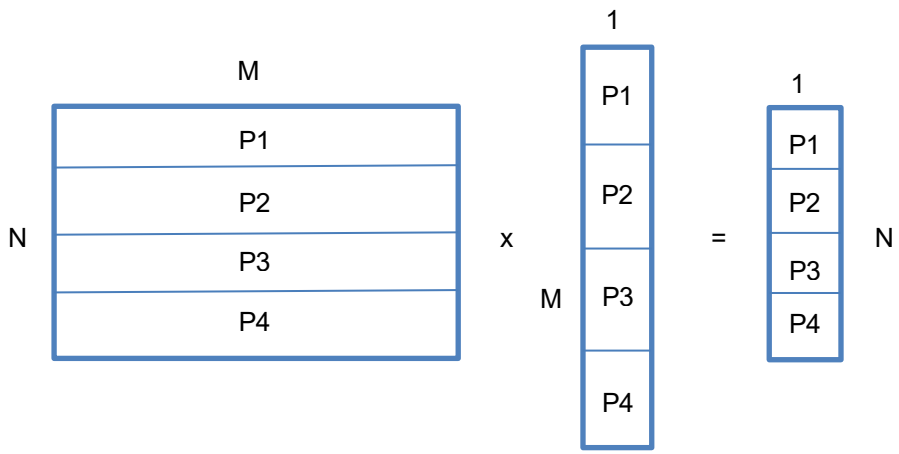
Responde a estas preguntas:

A. ¿Es necesario poner una barrera después del *broadcast* para asegurarse que todos tienen los datos?

B. ¿Podemos hacer el scanf en todos los procesos?

Ejercicio 2. Recolección (*gather*)

Se trata en este punto de realizar en paralelo el producto de una matriz por un vector utilizando *gather*. La matriz y el vector son divididos por bloques de filas (paneles) entre los procesos. Para simplificar se supone que las dimensiones de la matriz son múltiplos del número de procesos. El vector es pasado a cada proceso con el *gather*. Se te da el programa secuencial para que compruebes los resultados. Piensa que operación usarías para que todos tuvieran la solución en cada proceso.



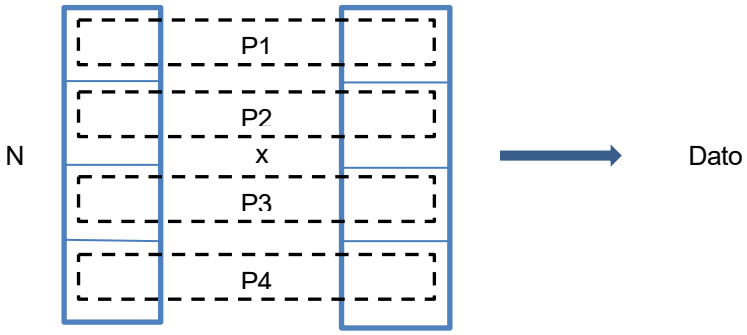
Ejercicio 3. Dispersión (scatter)

Se te da un programa secuencial que escribe 100 dobles en un fichero cuyo nombre se pasa como argumento. Realiza un programa MPI en el que el proceso raíz toma los datos de ese fichero (el nombre es pasado por cabecera) y los reparte (scatter) entre los N procesos (N debe ser un divisor de 100), que los pintarán en pantalla.

Ejercicio 4. Reducción (reduction)

A. Adapta el programa de los trapezoides para que el proceso raíz haga una "reduction" de tipo suma de los datos del resto de procesadores.

B. Realiza el producto escalar (dot product) de forma paralela de dos vectores de tamaño N entre P procesadores (N >P). N no es necesariamente múltiplo de P.



C. Y si todos necesitan el resultado ¿cómo lo cambiarías?

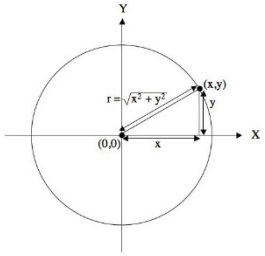
D. Como curiosidad, reproduce el cálculo factorial con una operación de reducción.

Ejercicio 5. Otras operaciones de reducción (opcional).

A. Calcula el valor de π con el método de los trapezoides usando la función:

$$\int_{-1}^{1} \sqrt{1-x^2}\,dx = \arcsin 1 = \frac{\pi}{2}$$

B. Ahora calcúlalo con el método de Montecarlo. Se trata de generar puntos aleatorios en un cuadrante de radio 1 y ver cuántos de esos puntos están dentro del círculo usando la fórmula del círculo: $X^2 + Y^2 < 1$ (Pitágoras). El master dará un número N de aleatorios a cada proceso MPI que lo hará por su cuenta y le dará al master el número obtenido de puntos dentro para que este calcule el resultado.



Π = Puntos dentro / N (para el cuadrante de 1 o por 4 para el total).

Ejercicio 6. Operaciones All to all

A. Reescribe el ejercicio 2 ¿podrías cambiar el bucle de gather por una única operación? Haz que todos los procesadores tengan la solución también.

B. Se tienen 4 vagones representados por cuatro procesos. Cada vagón tiene N (4) pasajeros representados por un array de N elementos, donde cada elemento es una estructura número/nombre. El número representa la clase de billete (de 1 a 4) y ese número se genera aleatoriamente de forma no repetida. Por un error de etiquetado, se han cambiado las clases por los asientos. En una parada de estación viene el revisor y reparte a los pasajeros de acuerdo a su tipo de billete, antes estaban sentado en cada vagón por orden de clase, ahora se trata de colocarlos en los vagones de acuerdo a su clase: pasajeros de la clase 1 (lujo) en el vagón 0... Estos "nuevos" pasajeros están representados por otro array de 4. ¿Cómo lo realizarías la ordenación con una llama MPI? Ten en cuenta que por ahora, para transmitir un array de estructuras lo tienes que convertir ("castear") a un array de bytes y transmitirlo y en la recepción lo mismo pero en orden diferente.

   En el tema siguiente verás cómo solucionar este problema de otra manera.