

PSD

Ignacio Bartolomé Tabanera
47284982-H

Roberto Morgado Luengo
02289149-M

Rubén Soto Ponce
05315494-X

Práctica 2

MPI

Índice

1. Implementación de multiplicación de matrices

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3
- Ejercicio 4

2. Implementación de producto interior de dos vectores

- Ejercicio 6
- Ejercicio 7
- Ejercicio 8(Opcional)

DESCRIPCIÓN GENERAL

Al ejecutar nuestra práctica con el comando `mpiexec` se le pasa como argumento el tamaño `N` de la dimensión de tanto la matriz como de la del vector en todos los ejercicios.

Ejemplo:

`mpiexec -n (Numero de procesos) ejercicio1 (Dimensión)`

MULTIPLICACIÓN DE MATRICES

EJERCICIO 1

Hemos implementado un programa secuencial que se encarga de la multiplicación de dos matrices cuadradas del tamaño indicado por parámetro (`N`).

Nos ha salido un tiempo de ejecución de 0,000004 segundos

EJERCICIO 2

Hemos paralelizado el programa anterior de forma que usando **MPI_BCAST** y **MPI_SEND** y **MPI_RECV**.

Inicializamos las matrices con los valores que deseamos.

El coordinador envía los trozos la matriz `A` a cada trabajador (**MPI_SEND**) dependiendo del número de trabajadores y el tamaño de la matriz (`numElements`), posteriormente cada trabajador recibe el trozo correspondiente de la matriz `A` (**MPI_RECV**).

El coordinador envía la matriz `B` entera a todos (**MPI_BCAST**).

Luego cada trabajador opera con las matrices que tiene y se encarga de enviar el resultado obtenido al coordinador (**MPI_SEND**) y finalmente el coordinador recibe todos los trozos con la operación ya calculada (**MPI_RECV**). Una vez tenemos el resultado lo mostramos por pantalla

Con dos procesos y 4 de dimensión el tiempo de ejecución es: 0,000035 segundos

EJERCICIO 3

Inicializamos las matrices con los valores que deseamos.

El coordinador envía los trozos la matriz A a cada trabajador (**MPI_ISEND**) dependiendo del número de trabajadores y el tamaño de la matriz (**numElements**), posteriormente cada trabajador recibe el trozo correspondiente de la matriz A (**MPI_RECV**).

El coordinador envía la matriz B entera a todos para tener una copia (**MPI_BCAST**).

Luego cada trabajador opera con las matrices que tiene y envía el resultado obtenido al coordinador (**MPI_SEND**) y luego éste recibe todos los trozos con la operación ya calculada previamente (**MPI_IRECV**). Una vez obtenido los resultados los mostramos por pantalla.

La diferencia es que en este caso el coordinador tiene que esperar a que los demás trabajadores hayan hecho el **MPI_ISEND** ya que lo hacemos asíncronamente (**MPI_WAIT**).

Todo esto se controla mediante la estructura **MPI_REQUEST** en la que se contienen las peticiones de los trabajadores.

Con dos procesos y 4 de dimensión es: 0,000023 segundos

EJERCICIO 4

Hemos paralelizado el programa anterior de forma que usando **MPI_BCAST** y **MPI_SCATTER** y **MPI_GATHER**.

A diferencia de los otros apartados en este el coordinador envía a cada trabajador el trozo correspondiente mediante **MPI_SCATTER** en lugar de hacer un bucle for para enviarle a cada uno el trozo que le corresponda de la matriz A se reparten los trozos solamente con esta sentencia.

Luego el coordinador envía a cada uno la matriz B mediante **MPI_BROADCAST**.

Posteriormente cada trabajador ejecuta las operaciones y finalmente el coordinador recoge los resultados con **MPI_GATHER** de todos los trabajadores con esta sentencia y lo muestra por pantalla.

Hay que tener en cuenta que los datos enviados a los procesos tienen que ser divisibles por el número de procesos para usar el **MPI_SCATTER** y **MPI_GATHER**.

Con dos procesos y dimensión 4 el tiempo de ejecución es: 0,000037 segundos

PRODUCTO INTERIOR DE VECTORES

EJERCICIO 6

Hemos realizado un programa secuencial que calcula el producto interior de dos vectores.

Inicializamos los vectores con unos valores escogidos, realizamos la multiplicación y calculamos la suma de los productos para obtener el resultado final.

Mostramos por pantalla los vectores y el resultado del producto escalar.

Con dos procesos y dimensión 4 el tiempo de ejecución es: 0,000052 segundos

EJERCICIO 7

Usamos dos vectores que vamos a multiplicar los cuales el coordinador los inicializa a unos determinados valores.

Luego el coordinador envía a los trabajadores los trozos del vector que les corresponden, es decir, el trozo del vector A que les corresponde y el trozo de B que le corresponde (**MPI_SCATTER**).

Después cada trabajador realiza la multiplicación y el coordinador con **MPI_REDUCE** realiza la suma de los resultados parciales obtenidos independientemente de cada trabajador y mostramos el resultado final de la suma que indica el producto escalar de los dos vectores.

Hay que tener en cuenta que los datos enviados a los procesos tienen que ser divisibles por el número de procesos para usar el **MPI_SCATTER** y **MPI_GATHER**.

Con dos procesos y dimensión 4 el tiempo de ejecución es: 0,000105 segundos

EJERCICIO 8 (OPCIONAL)

Usamos dos vectores que vamos a multiplicar los cuales el coordinador los inicializa a unos determinados valores.

Hay que tener en cuenta que el tamaño de los datos enviados a los trabajadores no tiene que ser divisible con el número de procesos. Para resolver esto usamos 2 nuevos vectores del tamaño de números de procesos que son **tamano[numnodes]** que guarda el tamaño de los datos que se envía a cada trabajador y **desplazamiento[numnodes]** que indica el desplazamiento que hay que añadir según el tamaño de los datos.

Luego el coordinador mediante dos **MPI_SCATTERV** envía a los trabajadores los trozos del vector que les corresponden, es decir, el trozo del vector A que les corresponde y el trozo de B que le corresponde.

Después cada trabajador realiza la multiplicación y el coordinador con **MPI_REDUCE** realiza la suma de los resultados parciales obtenidos independientemente de cada trabajador y mostramos el resultado final de la suma que indica el producto escalar de los dos vectores.