***PSD***

**Ignacio Bartolomé Tabanera**

**47284982-H**

**Roberto Morgado Luengo**

**02289149-M**

**Rubén Soto Ponce**

**05315494-X**

## 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

**DESCRIPCIÓN**

Al ejecutar nuestra practica con el comando mpiexec se le pasa como argumento el tamaño N de la dimensión de tanto la matriz como de la del vector.

EJ:

**mpiexec –n (Numero de procesos) ejercicio1 (Dimension)**

**EJERCICIO 1**

Hemos implementado un programa secuencial que se encarga de la multiplicación de dos matrices cuadras de tamaño según le pases por parámetro .

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 worker dependiendo del numero de workers y el tamaño de la matriz,(MPI\_SEND)posteriormente cada trabajador recibe el trozo correspondiente de la matriz A(MPI\_RECV).Luego el coordinador envía la matriz B entera a todos (MPI\_BCAST).

Luego cada worker opera con las matrices que tiene y el trabajador posteriormente envía el resultado obtenido al condinador(MPI\_SEND) y luego el coordinador recibe todos los trozos con la operación ya calculada(MPI\_RECV).Y posteriormente se muestra por pantalla.

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

**EJERCICIO 3**

Inicializamos las matrices con los valores que deseamos.

El coordinador envía los trozos la matriz A a cada worker dependiendo del numero de workers y el tamaño de la matriz,(MPI\_ISEND)posteriormente cada trabajador recibe el trozo correspondiente de la matriz A(MPI\_RECV).Luego el coordinador envía la matriz B entera a todos (MPI\_BCAST).

Luego cada worker opera con las matrices que tiene y el trabajador posteriormente envía el resultado obtenido al coordinador(MPI\_SEND) y luego el coordinador recibe todos los trozos con la operación ya calculada(MPI\_IRECV).Y posteriormente se muestra por pantalla.

La diferencia es que en este caso coordinador tiene que esperar a que los demás workers 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.

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 worker el trozo correspondiente mediante MPI\_SCATTER en lugar de hacer un for para enviarle a cada uno el trozo que le corresponda de la matriz A.

Luego se envía a cada uno la matriz B mediante MPI\_BROADCAST, posteriormente cada worker ejecuta las operaciones y finalmente el coordinador recoge los resultados con MPI\_GATHER y lo muestra por pantalla.

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

**EJERCICIO 5**

**EJERCICIO 6**

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

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

**EJERCICIO 7**

Tenemos dos vectores que vamos a multiplicar los cuales el coordinador los inicaliza a unos determinados valores.

Luego el coordinador mediante dos MPI\_SCATTER envía a los workers 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 worker realiza la multiplicación y el coordinador con MPI\_REDUCE realiza la suma de los resultados obtenidos independientemente de cada worker y los muestra por pantalla.

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

**EJERCICIO 8**

Tenemos dos vectores que vamos a multiplicar los cuales el coordinador los inicaliza a unos determinados valores.El numero de procesos en este caso no tiene porque ser divisible entre la dimensión de los vectores.

Luego el coordinador mediante dos MPI\_SCATTERV envía a los workers 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 con los

Después cada worker realiza la multiplicación y el coordinador con MPI\_REDUCE realiza la suma de los resultados obtenidos independientemente de cada worker y los muestra por pantalla.