

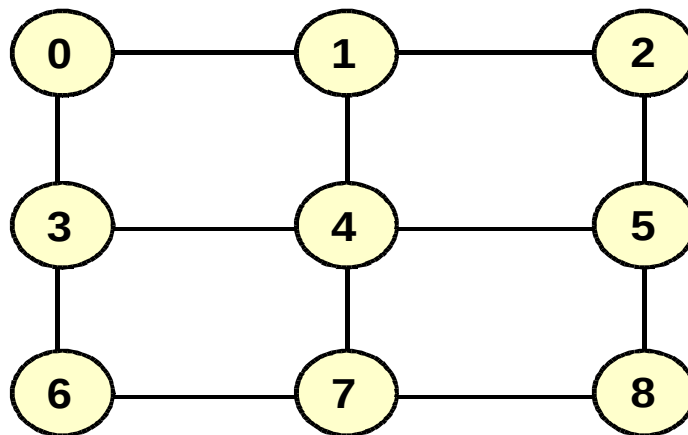
Relación de Ejercicios Tema 1. Programación Paralela 4º de Grado en Ingeniería Informática.

1. ¿Por qué el modelo de programación que se sigue al programar con MPI es independiente de la asignación?
2. Describir gráficamente una solución eficiente para realizar una operación de reducción global con 8 procesos (suma, máximo, producto, etc.) suponiendo dos casos: a) que la solución debe obtenerse sólo en una de las tareas, b) que la solución debe obtenerse en todas las tareas. Describir la estructura de comunicación así como las operaciones que realiza cada tarea y evaluar el tiempo de comunicación y computación que requiere el algoritmo.
3. Describir gráficamente cómo se podría implementar una operación colectiva tipo MPI_Gather siendo raíz el proceso 0 sobre un 4 procesadores conectados con un hipercubo de dimensión 2. Obtener una fórmula de tiempo de ejecución para esta operación en función del tamaño del bloque que aporta cada procesador N y los parámetros de comunicación t_s (latencia) y t_w (ancho de banda) de la arquitectura. Intentar extender la fórmula para un hipercubo de cualquier dimensión.
4. Describir gráficamente cómo se podría implementar una operación colectiva MPI_Gather siendo raíz el proceso 0 en un árbol binario con 7 procesadores asumiendo que hay que cada procesador aporta un subvector de N elementos enteros. Obtener una fórmula de tiempo de ejecución para esta operación en función del tamaño del bloque que aporta cada procesador N y los parámetros de comunicación t_s (latencia) y t_w (ancho de banda suponiendo que cada entero ocupa una palabra de memoria) de la arquitectura.
5. Describir un algoritmo basado en hipercubo para resolver el problema de encontrar todos los ceros de una lista de enteros. Ilustrar el algoritmo para el caso de $P=4$, $P=8$ y para 32 números. Evaluar el tiempo de computación y comunicación del algoritmo en base a parámetros de la arquitectura y el tamaño del problema.
6. Describir un algoritmo para resolver el problema de calcular si el número de ceros y de unos de una secuencia binaria es par o impar (tanto para el número de unos como para el número de ceros). Ilustrarlo con una secuencia de $N=21$ bits con $P=4$ y $P=8$. Suponer que se desea que el resultado se obtenga en todos los procesadores que intervienen en la computación. Modelar el tiempo de ejecución del algoritmo suponiendo que el número de procesadores es una potencia de 2 en función del tamaño de la secuencia N y los parámetros de comunicación t_s (latencia) y t_w (ancho de banda) de la arquitectura.

Relación de Ejercicios Tema 1. Programación Paralela 4º de Grado en Ingeniería Informática.

7. Describir gráficamente cómo se podría implementar de forma eficiente una operación tipo MPI_Gather sobre $P=9$ procesadores conectados entre sí con una topología de interconexión malla cuadrada 3×3 , asumiendo que el procesador 0 es el procesador raíz del Gather. (véase la figura de abajo). Para ello, mostrar en qué instante de tiempo se realizan los diferentes pasos de comunicación, etiquetando con instantes de tiempo (1,2,3,...) los enlaces donde se produce la comunicación incluyendo también el tamaño de los mensajes que se transfieren. Se supone que cada procesador mantiene inicialmente un vector de M enteros y que al final de la operación el procesador 0 mantendrá un vector de $9M$ enteros

- Obtener una fórmula de tiempo de ejecución para esta operación en función de los parámetros de comunicación t_s (latencia) y t_w (ancho de banda para envío de enteros) de la arquitectura.
- Extender la fórmula para el caso en que P es el cuadrado de un entero positivo ($P=k^2$, con k entero positivo) y la malla tiene k filas y k columnas.



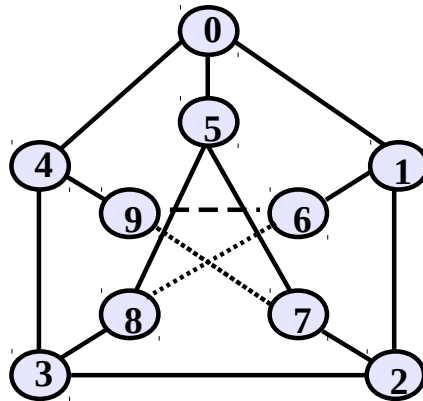
8. Se dispone de un algoritmo paralelo para calcular una aproximación de la integral de una función f en un intervalo $[a,b]$ sobre P procesadores. El algoritmo consta de los pasos siguientes:

- Un procesador distinguido raíz selecciona un entero N , múltiplo de P . Se supone que esta operación tiene coste 0. Este procesador raíz difunde el valor N a todos los demás utilizando un esquema de difusión en árbol.
- Cada procesador selecciona N/P subintervalos disjuntos del intervalo $[a,b]$ de tamaño $d=(b-a)/N$ y calcula una aproximación a la integral en dichos subintervalos, utilizando la regla del trapecio. Se supone que el costo asociado al cálculo de la integral en un subintervalo es t_{c1} . Cada procesador P_j calcula la suma local de las N/P aproximaciones calculadas. Podemos suponer que el costo de la suma aritmética de dos números reales es t_{c2} .
- La suma de las aproximaciones calculadas por cada procesador individual es obtenida en todos los procesadores mediante una operación de reducción todos a todos que sigue un esquema de hipercubo (se asume que P es una potencia de 2).

Suponiendo que todos los valores numéricos, sean enteros o reales, ocupan una palabra de memoria, obtener el tiempo de ejecución del algoritmo en función de N , P , t_s , t_w , t_{c1} y t_{c2} (t_s = costo de inicialización de envío, t_w = costo de transferencia por palabra de memoria).

Relación de Ejercicios Tema 1. Programación Paralela 4º de Grado en Ingeniería Informática.

9. Describir gráficamente cómo se podría implementar “de forma eficiente” una operación tipo MPI_Bcast sobre 10 procesadores conectados entre sí con una topología tipo “grafo de Petersen”, que se muestra en la figura de abajo, asumiendo que el procesador 5 es el procesador raíz del MPI_Bcast. Para ello, mostrar en qué instante de tiempo se realizan los diferentes pasos de comunicación, numerando con instantes de tiempo (1,2,3,...) los enlaces donde se produce la comunicación. Se supone que el procesador 5 mantiene inicialmente un vector de M enteros y que al final de la operación todos los procesadores mantendrán dicho vector en sus memorias locales. Obtener también una fórmula de tiempo de ejecución para esta operación de comunicación, en función de los parámetros de comunicación t_s (latencia) y t_w (ancho de banda para envío de enteros) de la arquitectura.



10. Se desea implementar una operación global paralela de suma de P vectores de enteros. Se supone que cada uno de los P procesos aporta un vector local de N enteros y que, tras ejecutar la operación, cada proceso obtiene como resultado un vector de N elementos que es igual a la suma de los vectores mantenidos localmente por cada proceso antes de ejecutar la operación.

- Describir cómo se puede implementar dicha operación colectiva sobre $P=2^k$ procesos (con k natural) conectados lógicamente con una estructura de comunicación hipercubo. Para ello, mostrar gráficamente con un ejemplo, cómo se puede implementar la operación para $P=4$ y $P=8$ procesos con $N=3$.
- Obtener el tiempo de ejecución del algoritmo descrito en función de N , P , t_s , t_w , t_c (t_c = costo de la suma de dos números enteros, t_s = costo de inicialización de envío, t_w = costo de transferencia por palabra de memoria que representa un entero). Obtener también una fórmula para la eficiencia del algoritmo.
- Cómo se podría realizar la operación utilizando una función de MPI. Describir cuál será la llamada exacta en MPI que permitiría realizar esta operación colectiva de suma distribuida de vectores de N elementos enteros.

Relación de Ejercicios Tema 1. Programación Paralela 4º de Grado en Ingeniería Informática.

11. Disponemos de un algoritmo paralelo para multiplicar una matriz de enteros A de tamaño $N \times N$ por un vector x de dimensión N obteniendo un vector $y = A \cdot x$. Se tiene una máquina con $P=7$ procesadores P_0, P_1, \dots, P_6 , conectados de acuerdo a una topología de árbol binario con 3 niveles. Se supone que N es un múltiplo de 7 y tanto la matriz A como el vector x se encuentran inicialmente en el procesador P_0 . El algoritmo procede de acuerdo a los siguientes pasos:

1. Desde P_0 se difunde la dimensión de la matriz (N) a todos los procesadores siguiendo un esquema de difusión en árbol binario.
2. El procesador P_0 reparte bloques de N/P filas consecutivas de la matriz A entre todos los procesadores. Para ello, se sigue un esquema de reparto en árbol, en el que el procesador P_0 se queda las primeras N/P filas de la matriz, parte el resto de la matriz en dos trozos iguales y los envía a sus dos hijos en el árbol, que harán exactamente lo mismo con la porción de matriz que les corresponde. Los nodos hojas se limitan a recibir un bloque de N/P filas de la matriz. Como resultado de este proceso, cada procesador P_i mantendrá N/P filas consecutivas de la matriz A .
3. P_0 difunde el vector x entre los procesadores siguiendo un esquema en árbol.
4. Cada procesador multiplica localmente su bloque de filas por el vector x , y obtiene un subvector con N/P elementos consecutivos del vector resultado y .
5. Finalmente P_0 recolecta los subvectores de y siguiendo un procedimiento inverso al seguido en 2 para los bloques de A (recorrido ascendente del árbol). Ahora los subbloques no son de N/P filas sino de N/P enteros (son subvectores del vector y).

a) Obtener una expresión para el tiempo de computación del algoritmo paralelo en función de N y t_c . (t_c = tiempo necesario para una suma y una multiplicación simple).

b) Obtener expresiones para las siguientes fracciones del tiempo de comunicación del algoritmo paralelo en función de N , t_s (tiempo de inicialización de envío) y t_w (tiempo de transferencia por cada entero enviado):

- Difusión de la dimensión de la matriz (N) entre los procesadores.
- Reparto equitativo de los bloques de filas de A entre los procesadores.
- Difusión del vector x entre los procesadores.
- Recolección de los subvectores del vector y .

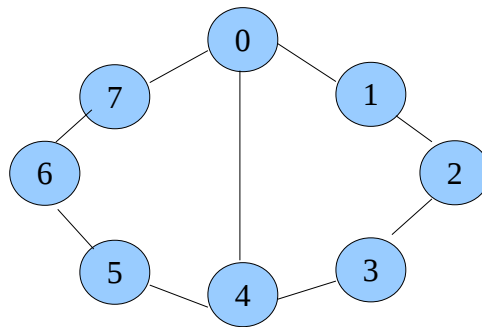
Relación de Ejercicios Tema 1. Programación Paralela

4º de Grado en Ingeniería Informática.

12. Describir gráficamente cómo se podría implementar de forma eficiente una operación tipo **MPI_Reduce** sobre **P=8** procesadores conectados entre sí con una topología de interconexión anillo con enlace central (véase la figura de abajo), asumiendo que el procesador 0 es el procesador raíz del **MPI_Reduce**, que la operación es una suma de reales en doble precisión y que cada procesador aporta un vector de **N** números reales. Para ello, mostrar en qué instante de tiempo se realizan los diferentes pasos de comunicación y computación, numerando con instantes de tiempo (1,2,3,...) los enlaces donde se produce la comunicación. Se supone que cada procesador mantiene inicialmente un vector de **N** números reales y que al final el procesador 0 mantendrá un vector resultado de sumar todos los vectores.

Suponer que un proceso I puede recibir un dato de un proceso J y, al mismo tiempo, enviar un dato “diferente” a otro proceso K (con $I \neq J \neq K$) usando enlaces diferentes.

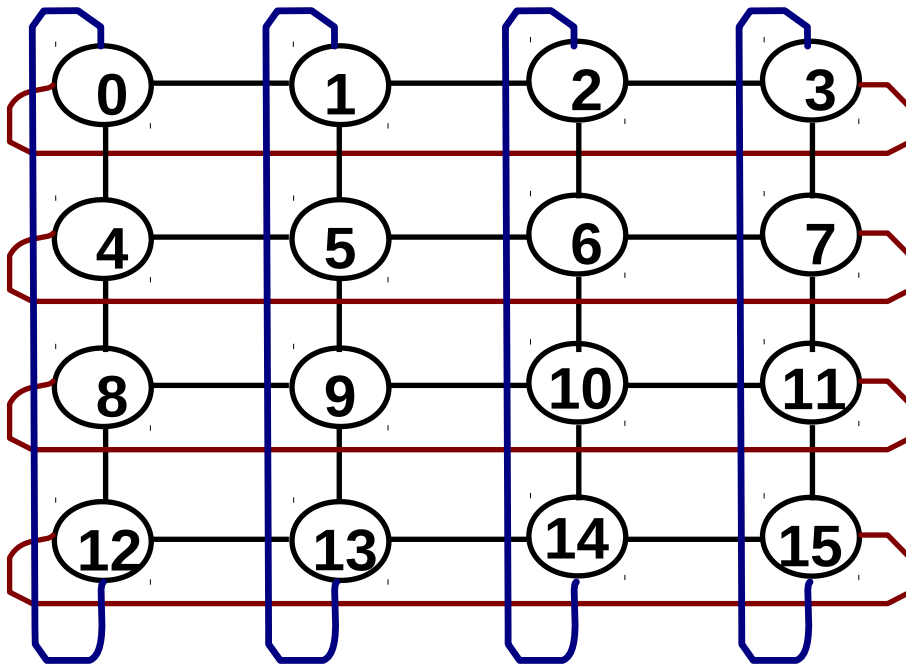
Describir gráficamente y textualmente la operación distribuida y obtener una fórmula de tiempo de ejecución para esta operación, sobre 8 procesadores, en función de **N** y los parámetros de comunicación **ts** (latencia), **tw** (ancho de banda para envío de números reales) y **tc** (tiempo dedicado a una suma de reales) de la plataforma usada.



Relación de Ejercicios Tema 1. Programación Paralela 4º de Grado en Ingeniería Informática.

13. Se necesita idear un algoritmo eficiente para repartir un vector \mathbf{X} de N elementos reales sobre $P=16$ procesadores, siendo N múltiplo de **16**. Se asume que todos los valores numéricos, sean enteros o reales, ocupan una palabra de memoria y que os 16 procesadores están conectados mediante una topología de interconexión malla toro 4×4 , tal como se muestra abajo:

Inicialmente el procesador 0 tiene el vector \mathbf{X} completo y se ha de distribuir por bloques de elementos consecutivos entre los **16** procesadores de tal forma que cada procesador obtendrá un subvector de $\mathbf{N/P}$ elementos. Se supone que se utiliza un esquema de reparto eficiente de los subvectores de \mathbf{X} siguiendo la topología de interconexión malla toro 4×4 . Ilustrar gráficamente cómo se realizaría este reparto y Obtener una fórmula de tiempo de ejecución paralela de esta fase en función de \mathbf{N} , \mathbf{P} , $\mathbf{t_s}$ y $\mathbf{t_w}$ ($\mathbf{t_s}$ = costo de inicialización de envío, $\mathbf{t_w}$ = costo de transferencia por palabra de memoria) y una descripción gráfica del procedimiento (describiendo los pasos de comunicación gráficamente en la figura de abajo) .



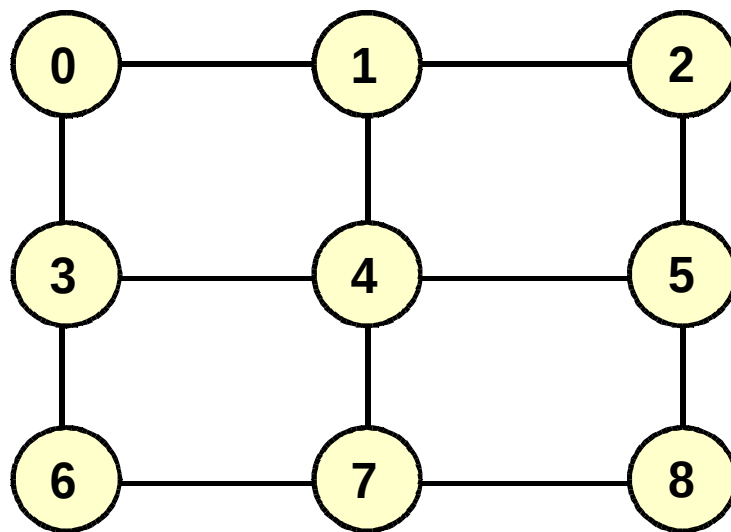
Relación de Ejercicios Tema 1. Programación Paralela

4º de Grado en Ingeniería Informática.

14. Se dispone de un algoritmo paralelo para calcular el producto escalar de dos vectores de N elementos reales X e Y sobre un número de procesadores $P=k^2$, con k entero positivo y N múltiplo de P . Se asume que todos los valores numéricos, sean enteros o reales, ocupan una palabra de memoria. Se desea modelar el tiempo de ejecución de cada una de las fases del algoritmo, que se ejecutan en este orden:

a) Inicialmente, asumimos que el procesador 0 tiene los vectores X e Y completos. Estos vectores se han de distribuir por bloques de elementos consecutivos entre los P procesadores de tal forma que cada procesador obtendrá dos subvectores de N/P elementos (uno del vector X y otro del vector Y). Se supone que se utiliza un esquema de reparto de los subvectores X e Y con una topología de interconexión malla cuadrada $k \times k$ y que el reparto de los subvectores se hace de forma conjunta para X e Y (no se hace primero para X y a continuación para Y , sino conjuntamente para mejorar la eficiencia).

Obtener una fórmula de tiempo de ejecución paralela de esta fase para $P=9$ (reparto de los vectores X y Y por bloques) en función de N , P , t_s y t_w (t_s = costo de inicialización de envío, t_w = costo de transferencia por palabra de memoria) y una descripción gráfica del procedimiento (describiendo los pasos de comunicación y el tamaño de los mensajes gráficamente en un diagrama como el de abajo).



b) Cada procesador P_i calcula el producto escalar de los subvectores de X e Y (de N/P elementos cada uno). Se supone que el costo asociado a la suma de dos reales es t_{c1} y el costo del producto de dos reales es t_{c2} . Obtener el tiempo de ejecución paralelo de este paso en función de N , P ($P=9$), t_{c1} y t_{c2} .

c) La suma de los valores obtenidos por cada procesador individual es obtenida en el procesador 0 mediante una operación de reducción que también sigue un esquema de malla cuadrada. Obtener una fórmula de tiempo de ejecución paralelo de este paso en función de P ($P=9$), t_s , t_w y t_{c1} , así como una descripción gráfica del procedimiento de reducción para $P=9$.