

Ejercicio 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 binario, 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 (en este caso sería un recorrido ascendente del árbol). Ahora los subbloques no son de N/P filas sino de N/P enteros (son subvectores del vector y).

Ejercicio 11.

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

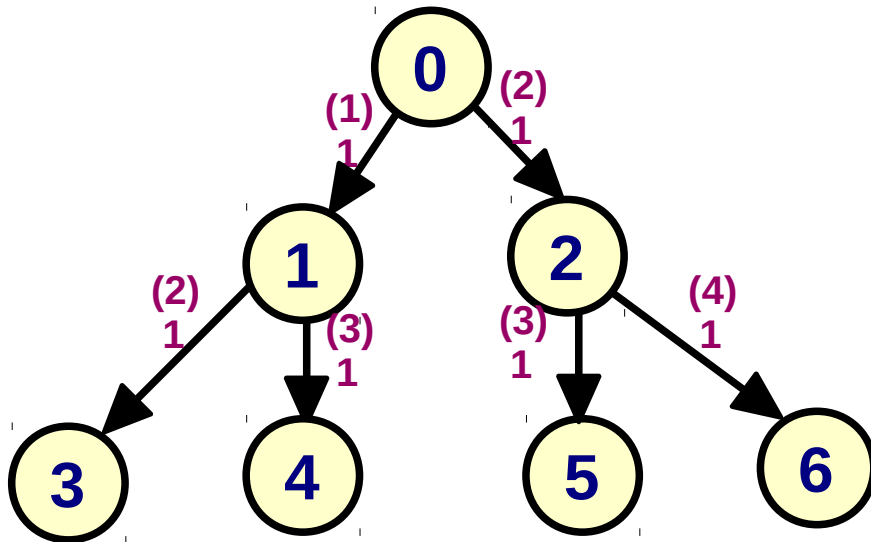
Toda la computación se realiza cuando cada procesador multiplica localmente su bloque de filas por el vector x, para obtener un subvector con N/P elementos consecutivos del vector resultado y.

Por lo tanto, se han de realizar N/P productos escalares del tipo fila_i (A)·x, donde cada uno requiere N productos y N-1 sumas, por lo que el coste de cada producto escalar se puede aproximar por N* tc. Como consecuencia:

$$T_{\text{comp.}} = N(N \cdot t_c) / P = N^2 P / t_c$$

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

b1) Difusión de la dimensión de la matriz (N) entre los procesadores.

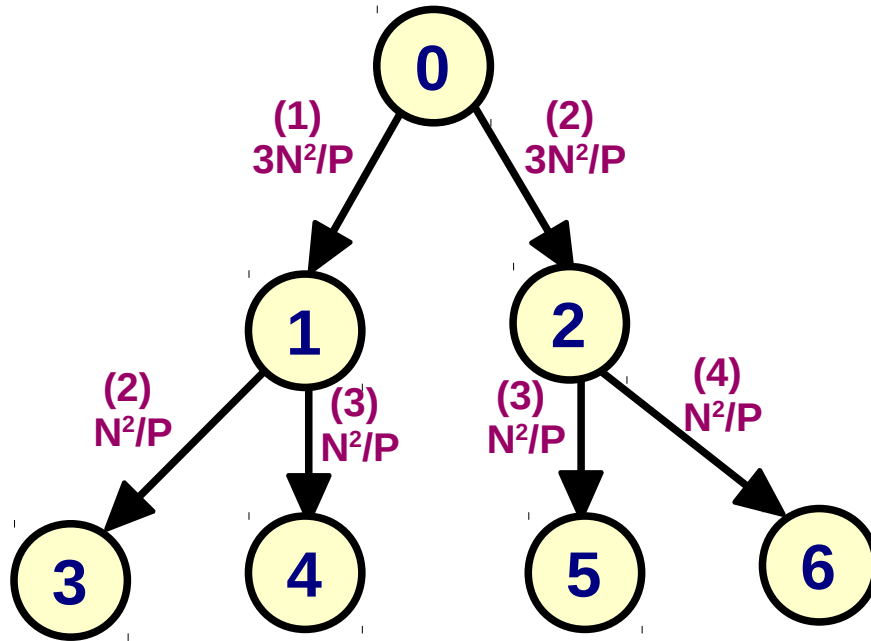


Para hacer un Broadcast en árbol binario de un elemento de forma eficiente, se recorren los nodos usando 4 pasos donde cada paso requiere el envío de un número entero. Por lo tanto:

$$T(\text{Broadcast}, N) = 4(ts + t_w)$$

Ejercicio 11.

b2) Reparto equitativo de los bloques de filas de A entre los procesadores.



Para hacer el reparto de los bloques de A en árbol binario, se hace en 4 etapas: las dos primeras consumen lo que se tarda en realizar dos envíos de $3N^2/P$ enteros ($P=7$) y las dos últimas lo necesario para 2 envíos de N^2/P .

$$T(\text{Scatter}, A) = 4t_s + 2t_w (3N^2/P + N^2/P) = 4t_s + 8t_w N^2/P$$

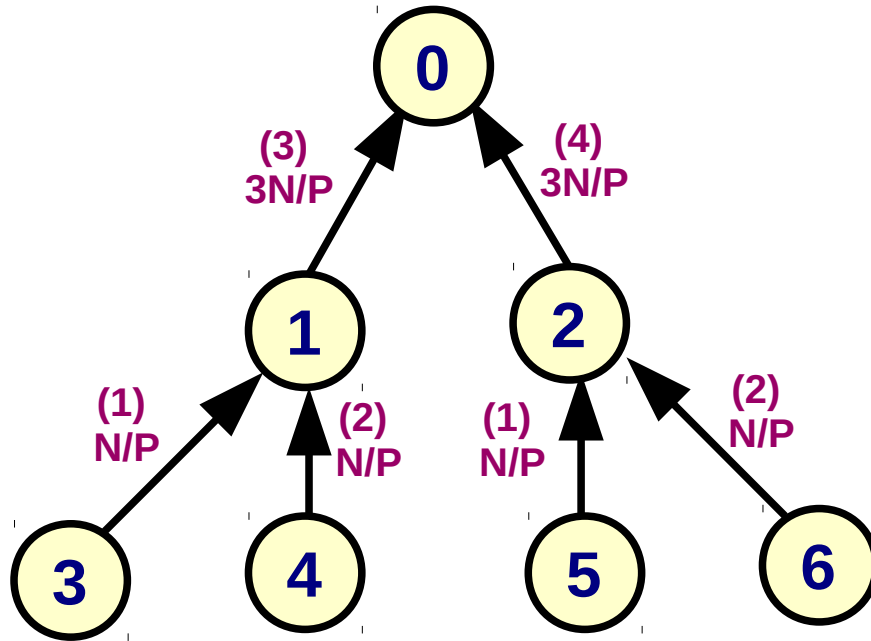
b3) Difusión del vector x entre los procesadores.

Se realiza con el mismo esquema que b1 pero teniendo en cuenta que se envían N enteros en cada paso en lugar de uno. Como consecuencia, se ha de multiplicar la componente t_w por N

$$T(\text{Broadcast}, x) = 4(t_s + Nt_w)$$

Ejercicio 11.

b4) Recolección de los subvectores del vector y .

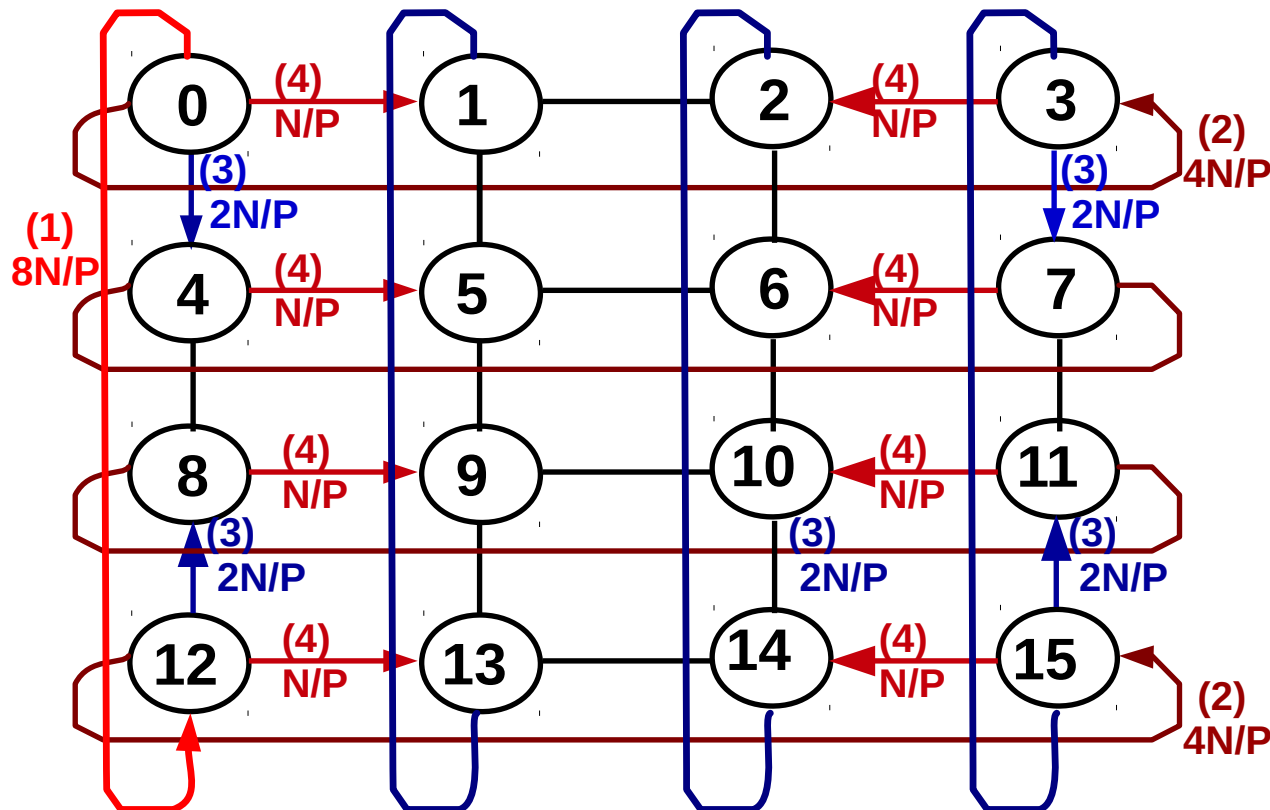


Para hacer el gather de los bloques de y en árbol binario, se hace en 4 etapas ascendentes: las dos primeras consumen lo que se tarda en realizar dos envíos de N/P enteros y las dos últimas lo necesario para 2 envíos de $3N/P$ enteros.

$$T(\text{Gather}, y) = 4t_s + 2t_w (N/P + 3N/P) = 4t_s + 8t_w N/P$$

Ejercicio 13.

Se necesita idear un algoritmo eficiente para repartir un vector 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 los 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 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 N/P elementos. Se supone que se utiliza un esquema de reparto eficiente de los subvectores de 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 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 gráficamente en la figura de abajo) .



El scatter se puede hacer en 4 etapas de envíos, empezando con un envío de 8 bloques (de N/P elementos cada uno) y dividiendo después el tamaño del mensaje en cada etapa hasta llegar a la etapa 4, donde se envía un solo bloque. Conforme avanzamos en el algoritmo, se consiguen más envíos en paralelo.

$$T(\text{Scatter}, X) = 4t_s + t_w (8 + 4 + 2 + 1)N/P$$

$$T(\text{Scatter}, X) = 4t_s + 15t_w N/P$$