

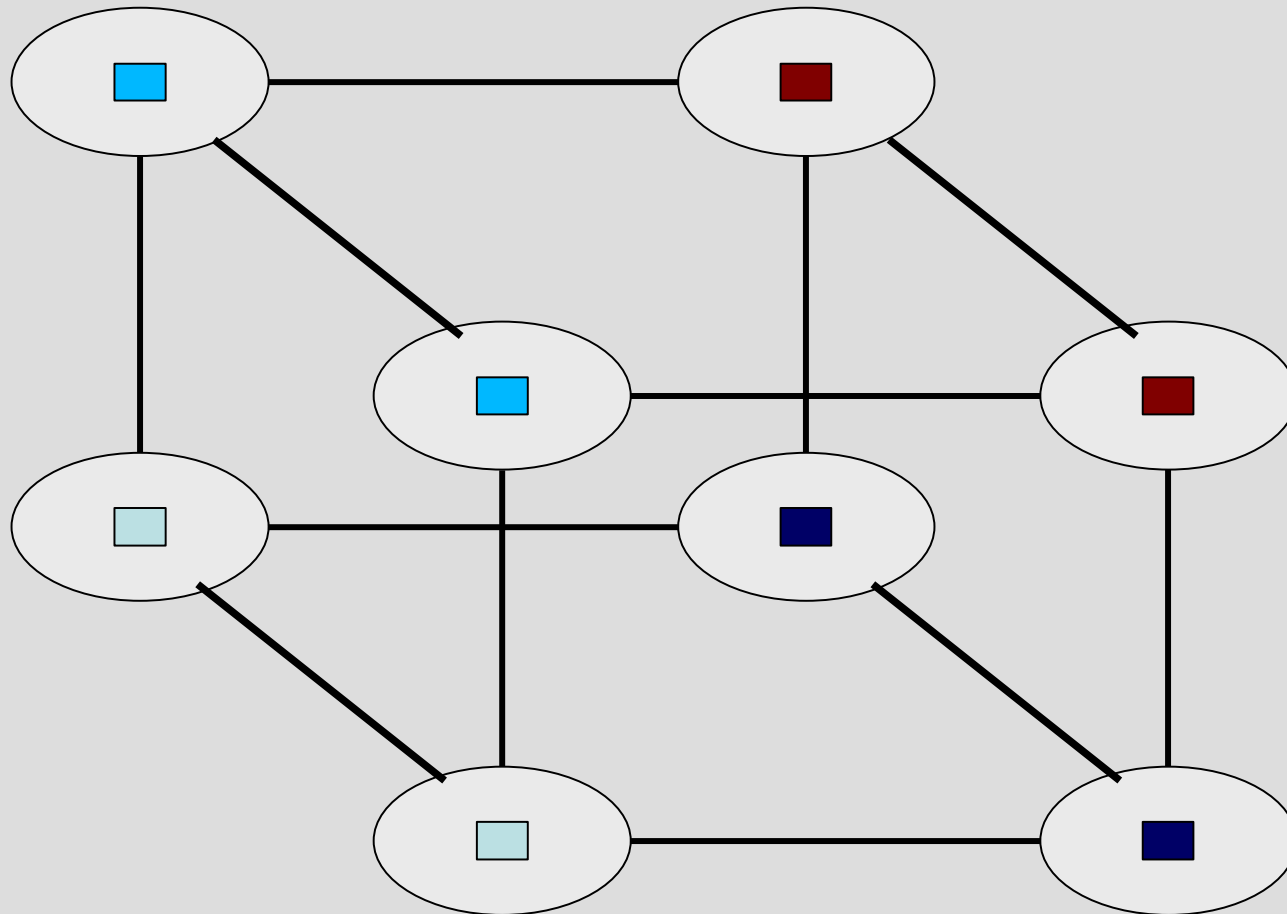
Ejercicios de Programación Distribuida y Paralela



*Depto. de Lenguajes y Sistemas Informáticos
Universidad de Granada*



TEMA 1. INTRODUCCIÓN



Ejercicio 1

Describir una solución eficiente para realizar una operación de reducción global con 2^n tareas (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.

Ejercicio 1. Solución

Estructura de comunicación: Hipercubo de dimensión n : $P=2^n$

Suponemos que cada tarea mantiene un bloque de elementos $B[1:N/P]$.

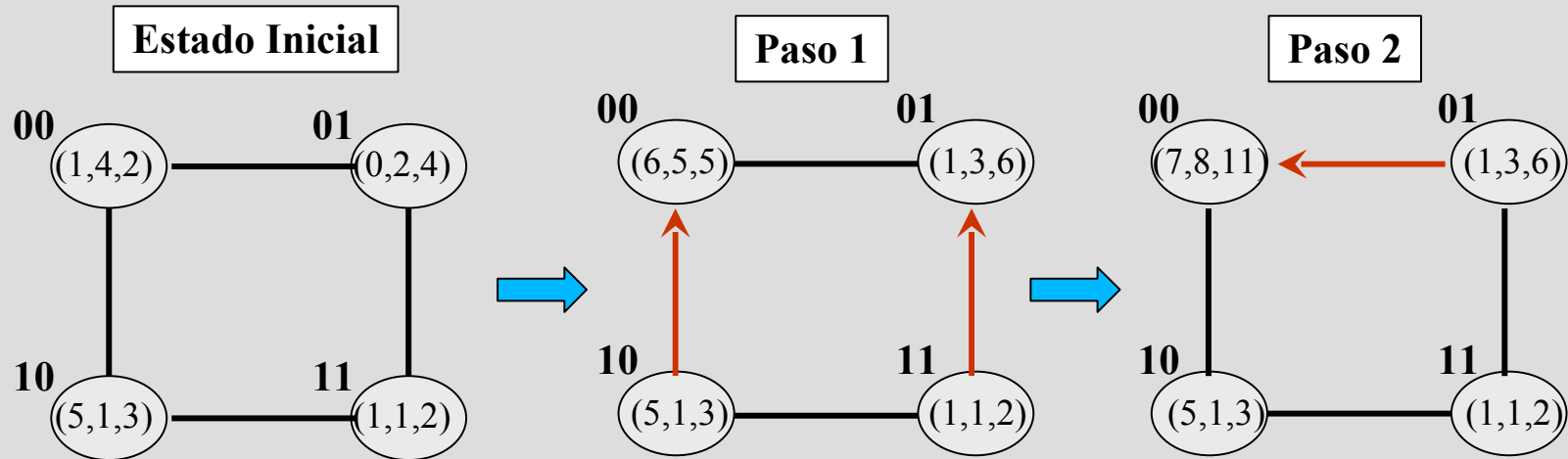
- $N_k(x)$ =Vecino del procesador x en la dimensión k del hipercubo.
- \otimes = operación a aplicar (suma, producto, mínimo, etc.).
- $\text{Digit}(k,x)$ =valor del dígito k -ésimo en repres. binaria del n° x .

```
a) Proceso x,  $0 \leq x < P$   
k = n-1;  
centinela=true;  
While (k>0 and centinela) {  
  If (digit(k,x)==1) {send(B, Nk(x));  
                     centinela=false;}  
  Else { receive (B2, Nk(x));  
         for i=1,N/P  
           {B[i] = B[i]  $\otimes$  B2[i];}}  
  k=k-1;}
```

```
b) Proceso x,  $0 \leq x < P$   
For k = n-1 downto 0 {  
  send(B, Nk(x))  
  receive (B2, Nk(x));  
  for i=1,N/P  
    {B[i] = B[i]  $\otimes$  B2[i];}  
}
```

Ejercicio 1. Solución

EJEMPLO caso a): $n=2$, operación suma con bloques de longitud $N=3$ y raíz P_0



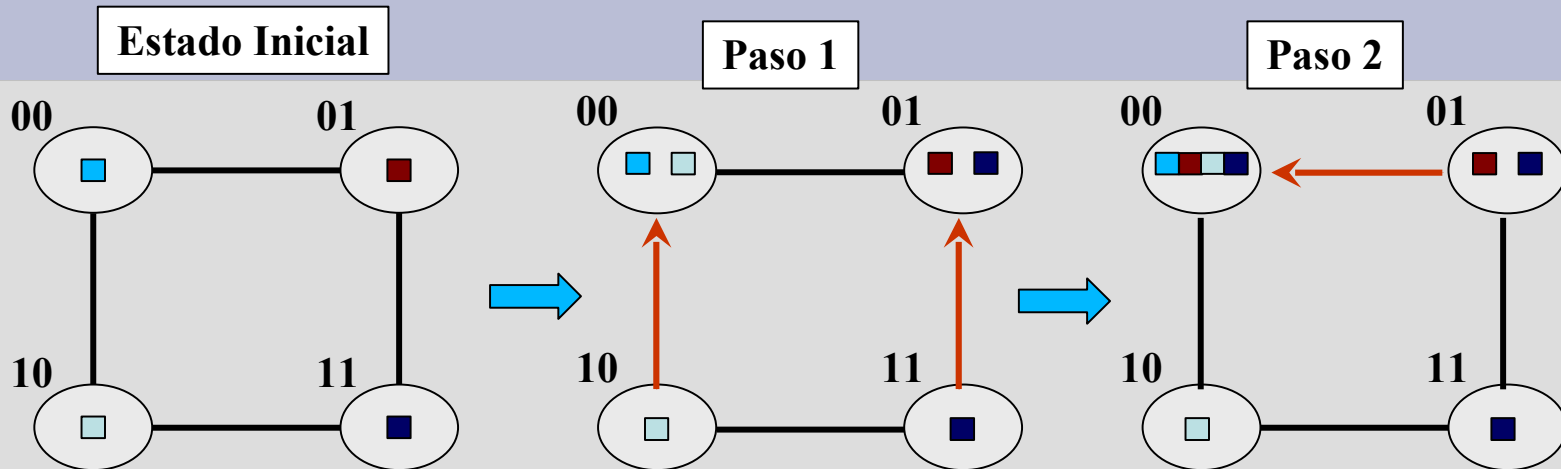
$$T_{reduce}(N, 2^2) = \boxed{t_s + Nt_w + t_c} + \boxed{t_s + Nt_w + t_c} = 2(t_s + Nt_w + t_c)$$

$$T_{reduce}(N, 2^n) = n(t_s + Nt_w + t_c) = \log P(t_s + Nt_w + t_c)$$

Ejercicio 2

Describir gráficamente cómo se podría implementar una operación colectiva MPI_Gather siendo raíz el proceso 0 en un hipercubo de dimensión 2 (con 4 procesadores). 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.

Ejercicio 2



Paso 1

$$T_{gather}(N,4) = t_s + Nt_w + t_s + 2Nt_w = 2t_s + 3Nt_w$$

Paso 2

$$T_{gather}(N,2^k) = kt_s + \sum_{i=0}^{k-1} 2^i Nt_w = kt_s + (2^k - 1)Nt_w$$

Ejercicio 5

Dada una lista de n números X_0, \dots, X_{n-1}

Se desea implementar la operación de prefijo que calcula todas las sumas parciales repartiendo los resultados en n procesos:

RESULTADO	PROCESADOR
x_0	P_0
$x_1 + x_0$	P_1
.....
$x_{n-1} + \dots x_1 + x_0$	P_{n-1}

Derivar una solución paralela a este problema siguiendo una estructura de hipercubo (suponemos de dimensión n).

Evaluar el tiempo de computación y comunicación del algoritmo.

Ejercicio 5. Solución

Proceso x , $0 \leq x < P$

resultado=mi_numero;

msg=resultado;

For k = 0 to n-1 {

Send (msg, Nk(x))

Receive (numero, Nk(x));

msg=msg+numero;

If (Nk(x)<x) resultado=resultado + numero; }

$$T_{prefix}(2^k) = \overbrace{k(t_s + t_w)}^{T_{comm}} + \overbrace{k t_c}^{T_{comp}} = \log P(t_s + t_w + t_c) \text{ Tiempo suma}$$