Ejercicio 1. Un programa tarda 40 s en ejecutarse en un multiprocesador. Durante un 20% de ese tiempo se ha ejecutado en cuatro procesadores; durante un 60%, en tres; y durante el 20% restante, en un procesador (consideramos que se ha distribuido la carga de trabajo por igual entre los procesadores que colaboran en la ejecución en cada momento, despreciamos sobrecarga). ¿Cuánto tiempo tardaría en ejecutarse el programa en un único procesador?¿Cuál es la ganancia en velocidad obtenida con respecto al tiempo de ejecución secuencial?¿Y la eficiencia?

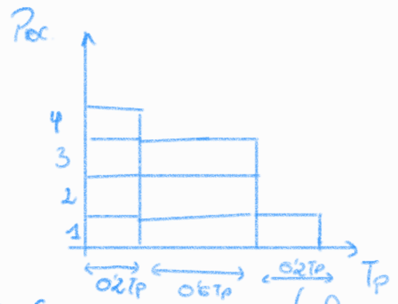
Datos.

Tiempo de cálculo

Tiempo de sobrecarga (en este caso despreciable)

·) $T_P(4) = 40s$

$T_P(p) = T_C(p) + T_O(p)$



El tamaño representa la carga de trabajo

Por tanto $T_S = 4 \times 0'2\,T_P + 3 \cdot 0'6\,T_P + 0'2\,T_P = 2'8\,T_P = 2'8 \cdot 4\,s = 112\,s$

Para obtener la ganancia. $S(4) = \frac{T_S}{T_P(p)} = \frac{2'8\,T_P}{T_P} = 2'8 < 4$ → número de procesadores

Es lógico pues $\max\{S(p)\} = p$ y $\min\{S(p)\} = 1$ → siempre que no haya sobrecarga.

Eficiencia.

$\mathcal{E}(p) = \frac{\text{Prestaciones}(p)}{p \cdot \text{Prestaciones}(1)} = \frac{S(p)}{p} = \frac{2'8}{4} = 0'7$

$\max\{\mathcal{E}(p)\} = 1$

$\min\{\mathcal{E}(p)\} = \frac{1}{p}$ { Siempre que no haya sobrecarga

Ejercicio 2. Un programa tarda 20 s en ejecutarse en un procesador P1, y requiere 30 s en otro procesador P2. Si se dispone de los dos procesadores para la ejecución del programa (despreciamos sobrecarga): $P_3 = 40s$

(a) ¿Qué tiempo tarda en ejecutarse el programa si la carga de trabajo se distribuye por igual entre los procesadores P1 y P2?

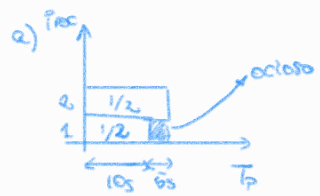
(b) ¿Qué distribución de carga entre los dos procesadores P1 y P2 permite el menor tiempo de ejecución utilizando los dos procesadores en paralelo ? ¿Cuál es este tiempo?

Datos

$T_S^{P_1} = 20s$

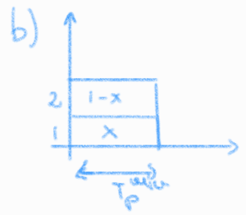
$T_S^{P_2} = 30s$

a)



$T_P(P_1,P_2)\left(\frac{1}{2},\frac{1}{2}\right) = \max\{10s, 15s\} = 15s$

b)



Asignación del trabajo

$T^{P_1}(x) = T^{P_2}(1-x)$

$20x = 30(1-x)$

$2x = 3 - 3x$

$5x = 3$

$\boxed{x = \frac{3}{5}}$ por tanto $\boxed{(1-x) = \frac{2}{5}}$

Cálculo del tiempo.

$\boxed{20s \cdot \frac{3}{5} = 12s}$

Ejercicio 5. En la Figura 1, se presenta el grafo de dependencia entre tareas para una aplicación. La figura muestra la fracción del tiempo de ejecución secuencial que la aplicación tarda en ejecutar grupos de tareas del grafo. Suponiendo un tiempo de ejecución secuencial de 60 s, que las tareas no se pueden dividir en tareas de menor granularidad y que el tiempo de comunicación es despreciable, obtener el tiempo de ejecución en paralelo y la ganancia en velocidad en un computador con:

↳ $T_C(p)$ despreciable

(a) 4 procesadores.
(b) 2 procesadores.

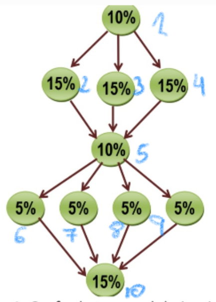
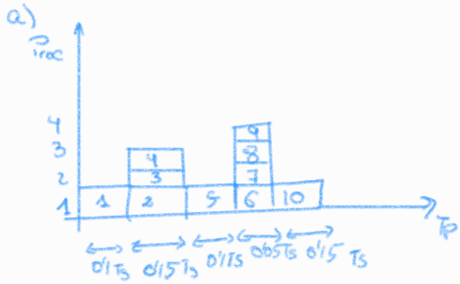


Figura 1. Grafo de tareas del ejercicio 5.

Datos:

$T_S = 60s$

a)

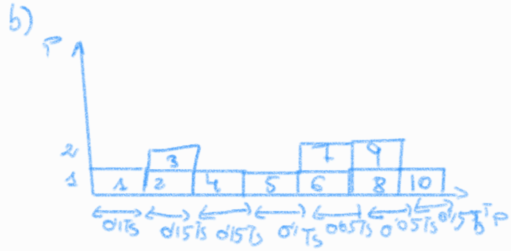


Por tanto, $T_P(4) = 0'1T_S + 0'15T_S + 0'1T_S + 0'05T_S + 0'15T_S = 0'55T_S = 0'32 + 0'01 s$

$S(4) = \frac{T_S}{T_P(4)} = \frac{T_S}{0'55} = \frac{1}{0'55} \approx 1'82 < 4$.

b)

Se hace la mejor asignación de tiempos posible



$T_P(2) = 3 \cdot 0'15T_S + 2 \cdot 0'05T_S + 2 \cdot 0'1T_S = 0'75T_S$

$S(4) = \frac{T_S}{0'75T_S} = \frac{1}{0'75} = \frac{4}{3} = 1'33 < 2$

Ejercicio 6. Un programa se ha conseguido dividir en 10 tareas. El orden de precedencia entre las tareas se muestra con el grafo dirigido de la Figura 2. La ejecución de estas tareas en un procesador supone un tiempo de 2 sg. El 10% de ese tiempo es debido a la ejecución de la tarea 1; el 15% a la ejecución de la tarea 2; otro 15% a la ejecución de 3; cada tarea 4, 5, 6 o 7 supone el 9%; un 8% supone la tarea 8; la tarea 9 un 10%; por último, la tarea 10 supone un 6%. Se dispone de una arquitectura con 8 procesadores para ejecutar la aplicación. Consideramos que el tiempo de comunicación se puede despreciar. **(a)** ¿Qué tiempo tarda en ejecutarse el programa en paralelo? **(b)** ¿Qué ganancia en velocidad se obtiene con respecto a su ejecución secuencial?.

$T_S = 2s$

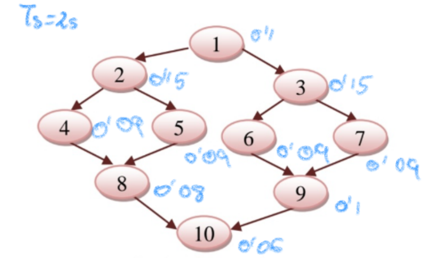


Figura 2. Grafo de dependencia entre tareas del ejercicio 6.

Mejor alternativa

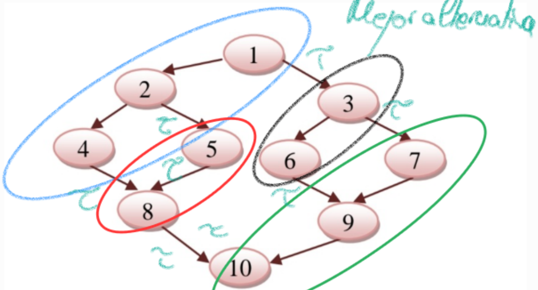


Figura 2. Grafo de dependencia entre tareas del ejercicio 6.

Para entender hay que hacer diagramas

Formas de asignación (cada color es un proc)

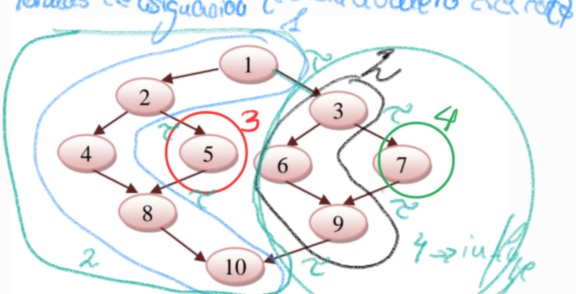
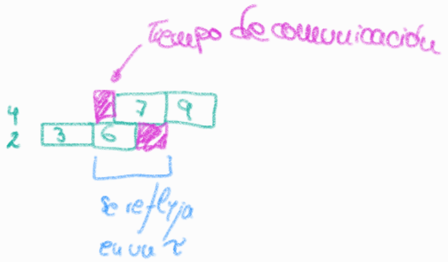


Figura 2. Grafo de dependencia entre tareas del ejercicio 6.

Es equivalente



Ejercicio 8. Supongamos que se va a ejecutar en paralelo la suma de *n* números en una arquitectura con *p* procesadores o cores (*p* y *n* potencias de dos) utilizando un grafo de dependencias en forma de árbol (divide y vencerás) para las tareas.

(a) Dibujar el grafo de dependencias entre tareas para *n* 16 y *p* igual a 8. Hacer una asignación de tareas a procesos.

(b) Obtener el tiempo de cálculo paralelo para cualquier n y p con n>p suponiendo que se tarda una unidad de tiempo en realizar una suma.

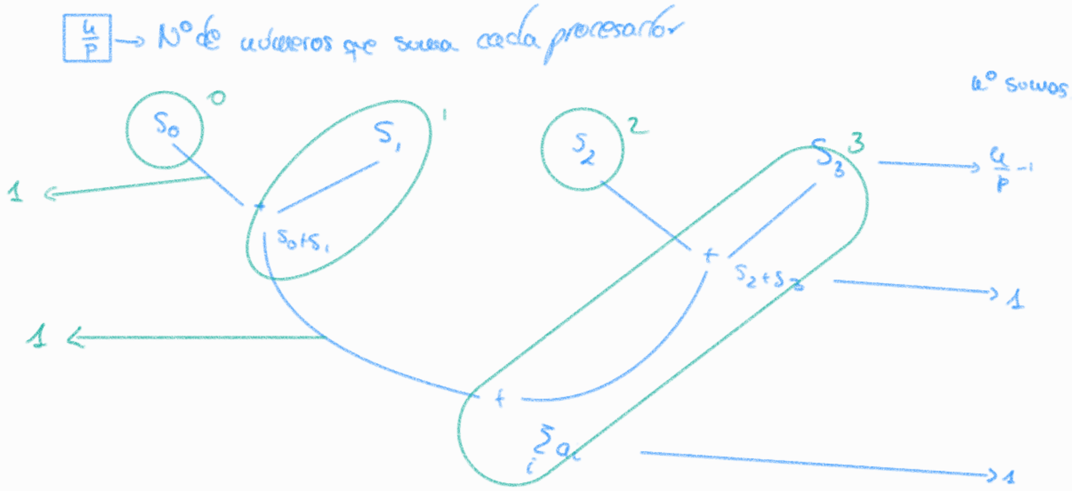
(c) Obtener el tiempo comunicación del algoritmo suponiendo (1) que las comunicaciones en un nivel del árbol se pueden realizar en paralelo en un número de unidades de tiempo igual al número de datos que recibe o envía un proceso en cada nivel del grafo de tareas (tenga en cuenta la asignación de tareas a procesos que ha considerado en el apartado (a)) y (2) que los procesadores que realizan las tareas de las hojas del árbol tienen acceso sin coste de comunicación a los datos que utilizan dichas tareas.

(d) Suponiendo que el tiempo de sobrecarga coincide con el tiempo de comunicación calculado en **(c)**, obtener la ganancia en prestaciones.

(e) Obtener el número de procesadores para el que se obtiene la máxima ganancia con *n* números.

a) Sumar n n^os en p procesadores

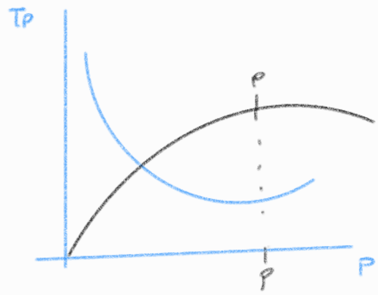
$\boxed{\frac{n}{p}} \rightarrow$ Nº de números que suma cada procesador



$T_p(p,n) = T_o(p,n) + T_{c,s}(p,n) = \left(\left(\frac{n}{p}-1\right) + \log_2(p)\right) + \log_2(p)$

$S(p,n) = \frac{T_s(n)}{T_p(n,p)} = \frac{n-1}{\left(\frac{n}{p}-1\right)+2\log_2(p)}$

$S(p,n) = R \Rightarrow \frac{n-1}{\frac{n}{p}-1} = p$

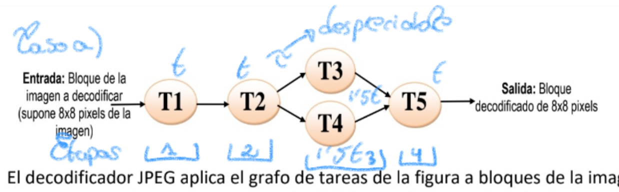


$T'(p) = 0$ → max → segunda derivada < 0

$T'(p) = 0$ → min → segunda derivada > 0

Ejercicio 9. Se va a paralelizar un decodificador JPEG en un multiprocesador. Se ha extraído para la aplicación el siguiente grafo de tareas que presenta una estructura segmentada (o de flujo de datos):

Caso a)



La tareas 1, 2 y 5 se ejecutan en un tiempo igual a t, mientras que las tareas 3 y 4 suponen 1,5t

$P_i \to T_i \quad i \in \{1,2,3,4,5\}$ → Asignación

El decodificador JPEG aplica el grafo de tareas de la figura a bloques de la imagen, cada uno de 8x8 píxeles. Si se procesa una imagen que se puede dividir en **n** bloques de 8x8 píxeles, a cada uno de esos **n** bloques se aplica el grafo de tareas de la figura. Obtenga la mayor ganancia en prestaciones que se puede conseguir paralelizando el decodificador JPEG en (suponga despreciable el tiempo de comunicación/sincronización): **(a)** 5 procesadores, y **(b)** 4 procesadores. En cualquier de los dos casos, la ganancia se tiene que calcular suponiendo que se procesa una imagen con un total de **n** bloques de 8x8 píxeles.

En un cauce segmentado

a) $T_p(5) = \underbrace{4'5t}_{B_1} + \underbrace{1'5t}_{B_2} + \underbrace{1'5t}_{B_3} + \dots + \underbrace{1'5t}_{B_n}$

$T_p(5) = T_{LI} + (n-1)\, T_{EMC}$

T_{LI} → T respuesta

T_{EMC} → Tiempo etapa más lenta

Tiempos.

B1: 0, t, 2t, 3'5t, 4'5t (etapas 1, 2, 3, 4)

B2: t, 2t, 3t, 3'5t, 5t, 6t { 1'5t

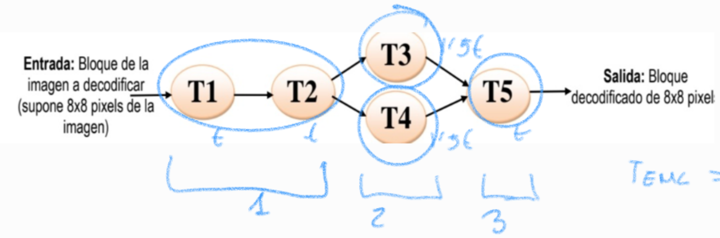
B3: 2t, 3t, 4t, 5t, 6'5t, 7'5t { 1'5t

$t_i \to$ tiempo T_i

$$S(5) = \frac{T_S(n)}{T_P(5,n)} = \frac{\left(\sum_{i=0}^{5} t_i\right)\cdot n}{T_{LI} + (n-1)\, T_{EMC}} \Rightarrow$$

$$\lim_{n\to\infty} S(5) = \frac{\sum_{i=1}^{5} t_i}{T_{EMC}} = \frac{6t}{1'5t} = \frac{6}{1'5} = \frac{6}{\frac{3}{2}} = \frac{12}{3} = 4$$

b)



$$S(4,n) = \lim_{n\to\infty} \frac{\sum_{i=1}^{4} t_i}{T_{EMC}} = \frac{6t}{2t} = 3$$

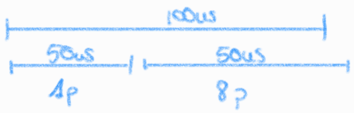
$T_{EMC} = 2t$

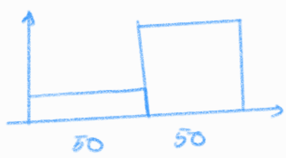
Ejercicio 3. ¿Cuál es fracción de código paralelo de un programa secuencial que, ejecutado en paralelo en 8 procesadores, tarda un tiempo de 100 ns, durante 50ns utiliza un único procesador y durante otros 50 ns utiliza 8 procesadores (distribuyendo la carga de trabajo por igual entre los procesadores)?

Datos

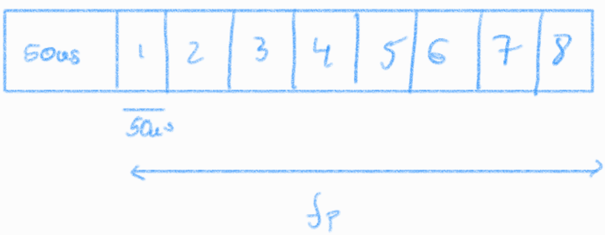
$T_p(8) = 100us$

$T_p(p) = T_c(p) + \cancel{T_o(p)}$ despreciable





Tiempo secuencial.



$\Rightarrow T_S = 50us + 8\cdot 50us = 9\cdot 50us$

$$f_P = \frac{8\cdot 50us}{T_S} = \frac{8\cdot 50us}{9\cdot 50us} = \frac{8}{9}$$

$$f_P = \frac{T_S(\text{paralelizado})}{T_S(\text{total})}$$

Ejercicio 4. Un 25% de un programa no se puede paralelizar, el resto se puede distribuir por igual entre cualquier número de procesadores. ¿Cuál es el máximo valor de ganancia de velocidad que se podría conseguir al paralelizarlo en *p* procesadores, y con infinitos? ¿A partir de cuál número de procesadores se podrían conseguir ganancias mayores o iguales que 2?

$f = 25\%$

$(1-f) = 75\%$

p procesadores

Vamos a aplicar la ley de Amdahl.

$$S(p) = \frac{p}{1+f(p-1)} = \frac{p}{1+\frac{p-1}{4}} = \frac{p}{\frac{4+p-1}{4}} = \frac{4p}{p+3}$$

Por tanto $\lim_{p\to\infty} S(p) = 4$ obteniendo la ganancia máxima

Ejercicio 7. Se quiere paralelizar el siguiente trozo de código:

```
{Cálculos antes del bucle}   t_antes
for( i=0; i<w; i++) {
      Código para i
}
{cálculos después del bucle}   T_despues
```

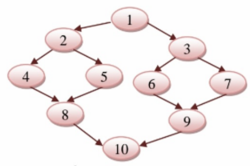


Figura 2. Grafo de dependencia entre tareas del ejercicio 6.

Los cálculos antes y después del bucle suponen un tiempo de t_1 y t_2, respectivamente. Una iteración del ciclo supone un tiempo t_i. En la ejecución paralela, la inicialización de *p* procesos supone un tiempo k_1p (k_1 constante), los procesos se comunican y se sincronizan, lo que supone un tiempo k_2p (k_2 constante); k_1p+k_2p constituyen la sobrecarga.

(a) Obtener una expresión para el tiempo de ejecución paralela del trozo de código en *p* procesadores (T_p).

(b) Obtener una expresión para la ganancia en velocidad de la ejecución paralela con respecto a una ejecución secuencial (S_p).

(c) ¿Tiene el tiempo T_p con respecto a **p** una característica lineal o puede presentar algún mínimo? ¿Por qué? En caso de presentar un mínimo, ¿para qué número de procesadores *p* se alcanza?

$$\left.\begin{array}{l} t_{antes} = t_1 \\ t_{despues} = t_2 \end{array}\right\} t$$

$t_{iteraci} = t_i$

$t_{inic}(p) = k_1 p$

$t_{sincro} = k_2 p$

$T_O(p) = k_1p + k_2p = p(k_1+k_2) = kp$

a) $T_p(p) = T_c(p) + T_O(p)$;

¿Cuántas iteraciones hace cada proceso? $\frac{w}{p}$

$T_{bucle} = \frac{w}{p} \cdot t_i$

$T_p(p) = t + \frac{w}{p} t_i + kp$

b) $S_p(p) = \frac{T_s}{T_p(p)} = \frac{t + w\, t_i + kp}{t + \frac{w}{p} t_i + kp}$

c) $T_p'(p) = \frac{-wt_i}{p^2} + k \Rightarrow T_p'(p) = 0 \Leftrightarrow \frac{-wt_i}{p^2} + k = 0 \Leftrightarrow \frac{wt_i}{p^2} = k \Leftrightarrow$

$T_p''(p) = \frac{+wt_i}{p^3}$

$\Leftrightarrow \boxed{p = \sqrt{\frac{wt_i}{k}}}$

$T_p''(p) = \frac{+k}{\sqrt{\frac{wt_i}{k}}} > 0 \Rightarrow$ luego un mínimo

Ejercicio 10. Se quiere implementar un programa paralelo para un multicomputador que calcule la siguiente expresión para cualquier x (es el polinomio de interpolación de Lagrange): $P(x) = \sum_{i=0}^{n} \left(b_i \cdot L_i(x)\right)$, donde

No sé ni por dónde empezar.

$$L_i(x) = \frac{(x-a_0)\cdot\ldots(x-a_{i-1})\cdot(x-a_{i+1})\cdot\ldots(x-a_n)}{k_i} = \frac{\prod_{\substack{j=0 \\ j\neq i}}^{n}(x-a_j)}{k_i} \qquad i = 0,1,\ldots,n$$

$$k_i = (a_i-a_0)\cdot\ldots(a_i-a_{i-1})\cdot(a_i-a_{i+1})\cdot\ldots(a_i-a_n) = \prod_{\substack{j=0 \\ j\neq i}}^{n}(a_i-a_j) \qquad i = 0,1,\ldots,n$$

Inicialmente k_i, a_i y b_i se encuentra en el nodo i y x en todos los nodos. Sólo se van a usar funciones de comunicación colectivas. Indique cuál es el número mínimo de funciones colectivas que se pueden usar, cuáles serían, en qué orden se utilizarían y para qué se usan en cada caso.

Los ejercicios 11, 12, 13, 14 constan de programación, se nos podrá preguntar sobre dichos ejercicios en clase?