Ejercicio 1

Datos

Tiempo Ejecución: 40s → 20% 4 procesadores; 60% 3 procesadores; 20% 1 procesador

Despreciamos overhead y suponemos que la carga se reparte de forma equitativa entre procesadores

Llamamos Tm al tiempo de ejecución del programa con varios procesadores (40s); entonces, eliminando el paralelismo y ejecutando con un solo procesador, el tiempo "secuencial" Ts sería:

$Ts = 0'2Tm + 4 \cdot 0'2Tm + 3 \cdot 0'6Tm$ porque el tiempo ejecutando varios threads en paralelo se multiplicaría por el número de threads.

Luego $Ts = (0'2 + 0'8 + 1'8)\,Tm = 2'8\,Tm$

Sea Sp la ganancia, calculada como $Sp = \frac{Ts}{Tm} = \frac{2'8\,Tm}{Tm} = \underline{2'8}$

La ganancia máxima posible sería de 4, luego la eficiencia del paralelismo es de

$2'8 \,\underline{|\,4}$ $0'7$

$Sp = \frac{Ts}{Tm} = 2'8$

$E = \frac{Sp}{4}$

Ejercicio 2

Datos:

Un programa A

Dos procesadores

P_1: ejecuta A en 20 s

P_2: ejecuta A en 30 s.

(a) Suponiendo que A se ejecuta con ambos procesadores al mismo tiempo y cada uno ejecuta la mitad de instrucciones ejecutaba él solo, tendríamos que el procesador P_1 ejecuta instrucciones durante 10 segundos y el P_2, durante 15 s.

Obviamente, solo se solaparían los primeros 10 segundos (suponiendo despreciable cualquier tipo de dependencia que impidiera ejecutar todas las instrucciones en paralelo).

En conclusión, el tiempo sería 15 s.

(b) Supongamos que x representa el porcentaje del programa que ejecuta P_1 e "y" representa el porcentaje que ejecuta P_2. Entonces, se tiene que cumplir que $x+y=1$ y además, el tiempo de ejecución total sería el máximo de "20x" y "30y"

$$\boxed{\text{MAX}}\ (20x,\ 30y)$$

$$x+y=1 \quad \text{con } x!=0\ y!=0$$

20 15
9/ 0'4

el punto en el que cambia el mínimo es:

$$20x = 30y$$
$$x+y=1 \rightarrow x=1-y$$

$$20(1-y)=30y$$
$$20-20y=30y$$
$$50y=20$$
$$y=20/50=2/5$$

$$x=1-0'4=\boxed{0'6}$$

$x=1$. $y=1-x$

Ejercicio 3 Para contestar calculemos el tiempo que tardaría en ejecutarse el programa secuencialmente: $50ns + 8 \cdot 50ns = 400ns + 50ns = 450ns$

$\begin{array}{r} 50 \\ \times 8 \\ \hline 400 \end{array}$

De los cuales 50ns no son paralelizables luego los otros 400ns son la parte paralela del programa:

$$\begin{array}{l} 450 - 100\% \\ 400 - x\% \end{array} \rightarrow x = \begin{array}{r|l} 40000 & 450 \\ 3600 & 8'8'88 \\ 02000 & \\ 200 & \end{array}$$

450 1
900 2
1350 3
1800 4
2250
2700
3150
3600
4050

84% paralelizable

Ejercicio 4

Sea T_S el tiempo secuencial de ejecución del programa. 0'75 sí.

$0'25 T_S$ no es paralelizable y

$\begin{array}{r|l} 0'75 & 3 \\ 15 & 0'25 \end{array}$

Suponiendo despreciable la sobrecarga u overhead:

$T_p(p)$ (tiempo paralelo para p procesadores) sería igual a:

$$T_p(p) = 0'25 T_S + \frac{0'75 T_S}{p} = \frac{0'25 T_S p + 0'75 T_S}{p} = \frac{T_S(0'25p + 0'75)}{p}$$

entonces

$$S(p) = \frac{T_S}{\frac{T_S(0'25p + 0'75)}{p}} = \frac{p}{0'25p + 0'75} = \frac{1}{0'25 + \frac{0'75}{p}}$$

Ganancia

Con infinitos: $\lim_{p \to \infty} \frac{1}{0'25 + \frac{0'75}{p}} = \frac{1}{0'25} = 4$

$$S(k) = 2 \iff \frac{1}{0'25 + \frac{0'75}{k}} = 2$$

$\begin{array}{r} 0'75 \\ \times 2 \\ \hline 150 \end{array}$

$$2\left(0'25 + \frac{0'75}{k}\right) = 1$$

$$0'5 + \frac{2 \cdot 0'75}{k} = 1 \rightarrow \frac{2 \cdot 0'75}{k} = 0'5$$

$$0'5k = 1'5 \rightarrow k = \frac{1'5}{0'5} = 3$$

ok sí!

$$k = 3 \qquad \frac{1}{0'25 + \frac{0'75}{3}} = \frac{1}{0'25 + 0'25} = \frac{1}{0'5} = 2$$

Ejercicio 5

$T_s = 60s$ (tiempo secuencial)

T_c despreciable (tiempo de comunicación)

Con 4 procesadores el tiempo sería:

$$T_p(4) = 0'1 T_s + \frac{0'15 T_s}{3} + 0'1 T_s + \frac{0'05 T_s}{4} + 0'15 T_s$$

$$T_p(4) = T_s(0'1 + 0'05 + 0'1 + 0'0150205 + 0'15) = \boxed{0'410205 T_s}$$

$$S_p(4) = \frac{T_s}{0'4102505 T_s} = \frac{1}{0'4102505} \simeq \boxed{2}$$

0'4102505 × 2 = 0'8205010

0'4102505 × 3 = 1'2307515

$$S_p(2) = \frac{T_s}{1'25 T_s}$$

Con 2 procesadores:

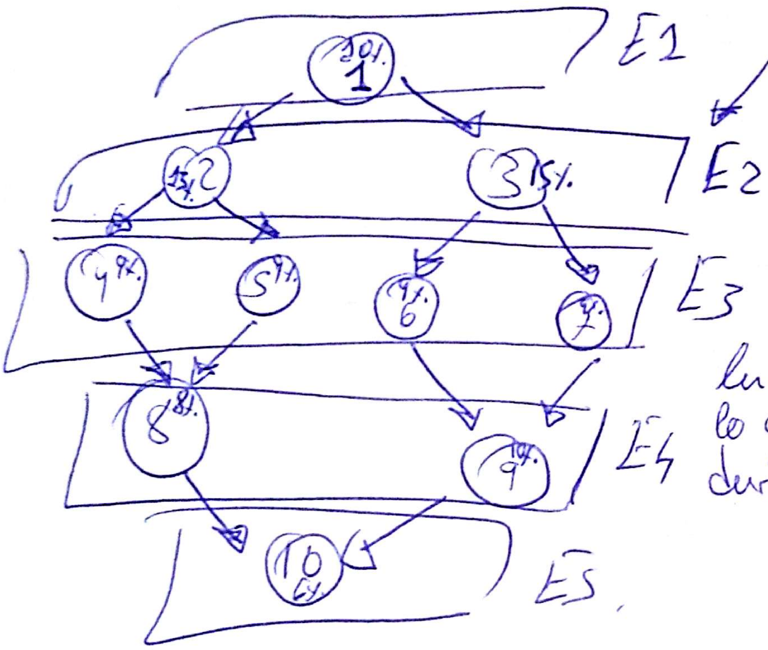
$$\left(0'1 + \frac{0'15}{2} + 0'15 + 0'1 + \frac{0'05}{2} + \frac{0'05}{2} + 0'15\right) T_s$$

$$0'1 + 0'7 + 0'15 + 0'1 + 0'05 + 0'15 = 1'25 T_s$$

$$\frac{1}{1'25} =$$

Gye6 Datos:

$T_S = 2seg.$ 8 procesadores.



Ambas terminan a la vez.

Todas terminan a la vez.

La tarea 9 es más lenta por lo que esta etapa dura lo que dure la ~~etapa 9~~ tarea 9.

~~$T_p(8) = T_S \left(0'1 + \frac{0'15}{2} + \frac{0'09}{4} + 0'1 \right.$~~

$(0'1 + 0'15 + 0'09 + 0'1 + 0'06) T_S$

Ejercicio 7

Sea $f = t_1 + t_2$ = tiempos ctes pre y post bucle, y

$k p = p(k_1 + k_2)$ = tiempo de sobrecarga u overhead

(a) $\boxed{T_p(p) = \frac{wt_i}{p} + kp + f}$

(b) El tiempo secuencial sería: $T_S = wt_i$

Luego $S_p = \frac{T_S}{T_p(p)} = \frac{wt_i}{\frac{wt_i}{p} + kp + f} = \frac{p\,wt_i}{wt_i + p^2 k + pf} = \frac{p}{1 + \frac{kp^2 + pf}{wt_i}}$

$$\boxed{S_p(p) = \frac{p}{1 + \frac{kp^2 + pf}{wt_i}}}$$

(c) Como $T_p'(p) = \frac{-wt_i}{p^2} + k = \frac{kp^2 - wt_i}{p^2}$

y se puede igualar a 0: $\frac{kp^2 - wt_i}{p^2} = 0 \iff wt_i = kp^2$

$\iff p^2 = \frac{wt_i}{k} \iff p = \sqrt{\frac{wt_i}{k}}$

y dado que w, t_i y k son positivos, p tendría un valor positivo posible, y dado que

$T_p''(p) = \frac{2p\,wt_i}{p^4} = \frac{2wt_i}{p^3}$ (que será positivo por ser los parámetros positivos)

podemos afirmar que la función $T_p(p)$ tiene un mínimo.

Ahora bien como p debe tomar valores enteros, ese mínimo podría darse en un número no válido de procesadores y habría que aproximarlo a un valor válido

ⓒ para 2^x números hay x etapas de las cuales se paralelizan completamente $y+1$ con $p=2^y$

Las no paralelizables se pueden ordenar de forma que no requieran comunicaciones, por lo que habría que comunicar las $y+1$ etapas paralelizables, salvo la primera de ellas, en la que todos los procesos tienen los datos a sumar, por lo que tendríamos y transferencias, cada una con el doble de datos que la anterior, empezando por una transmisión (que sería la parte entre la final y la anterior), luego:

$$g(n,p) = g(2^x, 2^y) = 1 + \sum_{i=1}^{y-1} 2^i = 2^y - 2 + 1 = 2^y - 1$$

tiempo de comunicación.

ⓓ $T_S = f(n,1) = f(2^x, 2^0) = 2^x - 1$

$$T_P(n,p) = \underbrace{2^{x-y} + y - 1}_{\text{cálculo}} + \underbrace{2^y - 1}_{\text{comunicación}} = 2^{x-y} + y + 2^y - 2$$

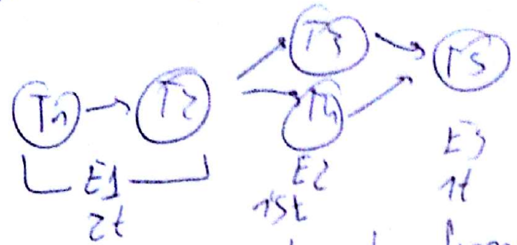
entonces $$S(n,p) = \frac{2^x - 1}{2^{x-y} + y + 2^y - 2}$$

para $n=16$, $p=8$ sería:

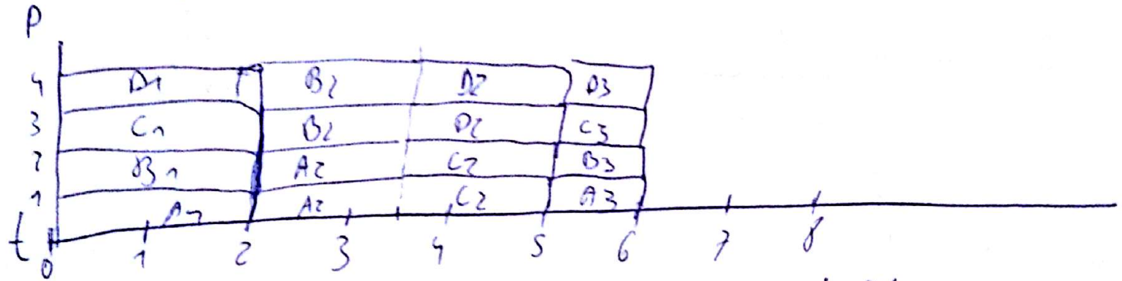
$$S(16,8) = \frac{2^4 - 1}{2 + 3 + 2^3 - 2} = \frac{16-1}{11} = \frac{15}{11} = 1'3636\ldots$$

Ejercicio 1

b) Con cuatro procesadores, sean A, B, C, D 4 bloques distintos y



Podemos paralelizar de esta forma:



De forma que en 6t se paralelizan 4 bloques

es decir, si secuencialmente un bloque se ~~procesa~~ decodificaría en 6t; con 4 procesadores paralelizamos cuatro bloques en 6t

luego tenemos una ganancia perfecta igual al número de procesadores $Sp(n) = 4$ porque para decodificar n ~~procesadores~~ bloques, si $n = 4x$

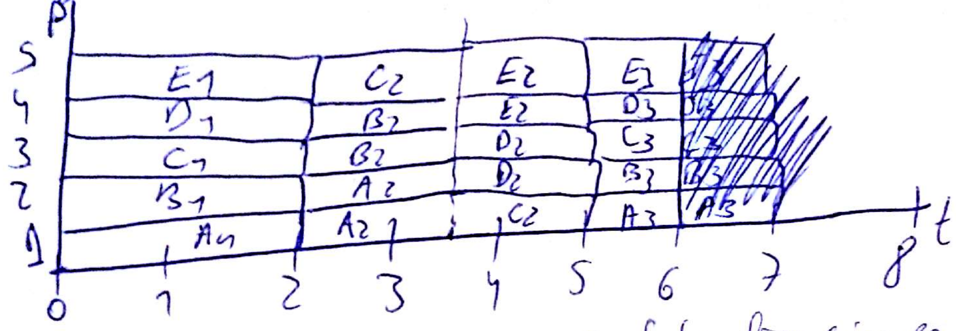
$Ts(n) = x4 \cdot 6t$; $Tp(n) = \frac{x4 \cdot 6t}{4} = 6tx$

$$Sp = \frac{4x6t}{x6t} = 4$$

luego ~~$Sp(n) = \frac{x4 \cdot 6t}{6t}$~~

obviamente si n no es un múltiplo de cuatro el resultado no sería óptimo porque habría algunos bloques ~~que no se para~~ (si fueran 3 o (entre 1 y 3)) que se ejecutarían en 6t; de cualquier modo la ganancia máxima seguiría en 4'5t; siendo 4.

Ejercicio 7 a)



Luego podríamos ejecutar 5 decodificaciones en 6t

entonces, si tenemos n = 5x bloques y secuencialmente se ejecuta un bloque en 6t, y podemos ejecutar 5 bloques en 6t, podemos deducir que tenemos una ganancia de 5 (aprox; si n != 5x entonces habría entre 1 y 4 bloques que no ~~sera~~ aprovecharían el paralelismo al máximo).