Solución del exámen de **Arquitecturas Avanzadas**

Prof. Fernando Pardo

1 Teoría

1.1 Ejercicio 1

El primer apartado es muy simple, ya que coincide con la teoría vista en clase:

$$T = R \max\{M-k, k\} + C(M-k)k$$

Para la segunda parte el que ejecuta la tarea α veces más rápido tardará α veces menos tiempo en ejecutar las tareas enconmendadas. Las comunicaciones las suponemos externas al procesador por lo que no se ven afectadas (si se supone que el α modifica las comunicaciones se da también por válido). Supongamos que el procesador con $M-k$ tareas es el más rápido:

$$T = R \max\left\{\frac{M-k}{\alpha}, k\right\} + C(M-k)k$$

El valor óptimo en este caso se dará cuando los dos procesadores estén equilibrados y tarden lo mismo, es decir, cuando se cumpla que:

$$\frac{M-k}{\alpha} = k$$

despejando la k en este ecuación se obtienen las tareas en el procesador lento:

$$k_{lento} = \frac{M}{\alpha+1}$$

y como $k_{rapido} = M - k_{lento}$ las tareas en el más rápido serán por tanto:

$$k_{rapido} = M - \frac{M}{\alpha+1} = M\frac{\alpha}{\alpha+1}$$

Para el tercer apartado basta igualar el tiempo que se tarda el procesador rápido con el tiempo que se tarda con los dos suponiendo el reparto anterior. El rápido a solas tarda RM unidades de tiempo, y si hay reparto entonces el tiempo es $R\frac{M}{\alpha+1} + C\frac{M}{\alpha+1}\frac{\alpha M}{\alpha+1}$ que se puede simplificar a $R\frac{M}{\alpha+1} + CM^2\frac{\alpha}{(\alpha+1)^2}$. Por lo tanto, igualando ambos términos se obtiene:

$$\frac{RM}{\alpha} = \frac{RM}{\alpha+1} + CM^2\frac{\alpha}{(\alpha+1)^2}$$

Despejando R/C de la expresión anterior se obtiene que $R/C = M\alpha^2/(\alpha+1)$. Por lo tanto, la condición para que sea rentable paralelizar era que se cumpliese:

$$\frac{R}{C} > M\frac{\alpha^2}{\alpha+1}$$

donde en este caso la R se refiere al coste de las tareas en el **procesador más rápido**.

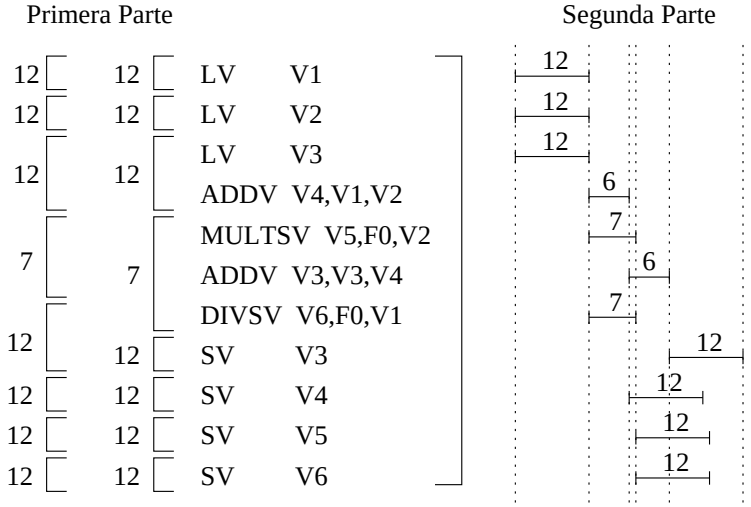
1.2 Ejercicio 7

El resultado es 16. Por la parte derecha se va sumando 1 cada vez al valor inicial. En la parte izquierda se van sumando de forma acumulada: $1 + 1 + 2 + 4 + 7 + 11 = 16$ y 16 es el que finalmente sale ya que al llegar a 5 el otro ramal se manda la etiqueta *Falso* que bloquea el resto de actuadores. La máquina se para puesto que no hay ningun actuador que tenga todas sus etiquetas.

2 Problemas

2.1 Problema 1

En la siguiente figura se muestran las posibles elecciones de convoyes tanto para el primer apartado como para el segundo. También se muestra el cálculo de los tiempos de arranque:



Arranque=12+12+12+7+12+12+12+12=91 Arranque=12+6+6+12=36

A la vista de la figura existen dos posibilidades para la primera parte del problema. En una posibilidad se está suponiendo que sólo hay un cauce de lectura para los registros escalares, y en la otra se extán suponiendo muchos; como en el problema no se han especificado cuántos había, ambas posibilidades son válidas, además dan lo mismo.

Con estas consideraciones y como hay 8 convoys:

$$T_n = \left\lceil \frac{n}{64} \right\rceil (15 + 91) + 8n$$

$15 + 91 = 106$ y quitando el redondeo por arriba para n grande se tiene:

$$T_n = \frac{106}{64} n + 8n + 106 = \frac{618}{64} n + 106 = 9.66n + 106$$

Con estas consideraciones el R_∞ se puede calcular mediante el límite de T_n/n con n tendiendo a infinito que da 9.66, con el numero de operaciones en coma flotante, que son 4, y con la frecuencia de reloj que son 200 MHz:

$$R_\infty = \frac{4 \cdot 200\, 10^6}{9.66} = 82'8 \text{ MFLOPS}$$

Si nos dicen que un procesador escalar tarda 25 ciclos en ejecutar las operaciones del vectorial por cada elemento, entonces el tiempo que tarda en ejecutar n instrucciones será:

$$T_n^e = 25n$$

El valor de N_v es tal que $T_n^e = T_n$ por lo que se igualan ambas expresiones y se despeja la n, para ello supondremos que $n < 64$:

$$25n = 106 + 8n$$

despejando n:

$$n = \frac{106}{25 - 8} = 6.23$$

Por lo tanto, el vector a partir del cual es más rentable usar un vectorial es $N_v = 7$.

Según las suposiciones del segundo apartado es únicamente necesario un convoy para ejecutar todo el código tal y como se mostraba al principio de la resolución de este problema. También ahí se mostraba el cálculo del tiempo de arranque que es de 36 ciclos de reloj. Con estas consideraciones el tiempo de ejecución en función del tamaño del vector es:

$$T_n = \left\lceil \frac{n}{64} \right\rceil (15 + 36) + 1n$$

Al igual que se hacía en la primera parte se ajusta el resultado suponiendo que n es grande, de esta manera hacemos desaparecer el redondeo superior:

$$T_n = \frac{51}{64}n + 51 + n = \frac{115}{64}n + 51 = 1.8n + 51$$

El límite es $\lim_{n\to\infty} T_n/n = 1.8$ con lo que el rendimiento asintótico queda:

$$R_\infty = \frac{4 \cdot 200\,10^6}{1.8} = 444.4 \text{ MFLOPS}$$

Para calcular $N_{1/2}$ supondremos que $N_{1/2} \leq 64$ y entonces utilizamos la fórmula del rendimiento en función de n y se iguala a $R_\infty/2$:

$$R_n = \frac{800n}{n + 51} = 222.2$$

$$(800 - 222.2)n = 51 \cdot 222.2$$

$$n = \frac{11332.2}{577.8} = 19.61$$

Redondeando este número al entero más próximo se obtiene el tamaño del vector tal que el rendimiento es aproximadamente la mitad del rendimiento máximo:

$$N_v = 20$$

2.2 Problema 2

La aceleración o *speed-up* es el tiempo con un procesador dividido por el tiempo con varios. El tiempo con un procesador es $T(1) = mR$ donde R es un factor que convierte carga en tiempo de ejecución. Con varios procesadores el tiempo tiene dos términos, por un lado mR/n que es lo que cuesta ejecutar las tareas entre todos los procesadores, y por el otro $R/4$ que es lo que ejecuta cada procesador para poner en marcha las tareas, o sea, $T(n) = mR/n + R/4$. De esta manera:

$$S = \frac{T(1)}{T(n)} = \frac{mR}{\frac{R}{4} + \frac{mR}{n}}$$

Multiplicando por 4 y dividiendo por R se obtiene la aceleración:

$$S = \frac{4mn}{n+m}$$

Para calcular el número de procesadores para el cual S empieza a caer se obtiene calculando el máximo de S por lo tanto hay que derivar e igualar a cero:

$$\frac{4m}{n+m} - \frac{4mn}{(n+m)^2} = 0$$

Realizando diversas simplificaciones se llega a que $m = 0$ que es imposible por lo que no existe un n a partir del cual el rendimiento decae. De hecho se puede calcular la aceleración asintótica:

$$S_{max} = \lim_{n\to\infty} S = \lim_{n\to\infty} \frac{4mn}{n+m} = 4m$$

Se trata entonces de una función siempre creciente pero que como mucho llega a $4m$ veces el rendimiento de un sólo procesador.

Otra suposición que se podría considerar aceptable a partir del enunciado, aunque no es estrictamente lo que se pedía, es que la sobrecarga es global a todos los procesadores y se suma para cada uno de ellos con lo cual en vez de ser $R/4$ sería $nR/4$. En este caso, y siguiendo las mismas pautas que en el caso anterior se tiene:

$$S = \frac{4mn}{n^2 + 4m}$$

En este caso se puede calcular el máximo y sí que se obtiene un valor de n a partir del cual la aceleración disminuye. Este valor es:

$$n = 2\sqrt{m}$$

También el límite en este caso cambia y se hace 0 cuando n tiende a infinito.