

A large, faint green graphic is centered in the background. It features a circular design with several curved arrows pointing outwards in different directions, creating a sense of dynamic movement or a network.

Ejercicios I

Arquitectura de Computadores

Dora Blanco Heras
Área de Arquitectura de Computadores
CITIUS
Universidad de Santiago de Compostela

Ejercicio 1

Estás diseñando un sistema para una aplicación que se va a ejecutar en tiempo real para cuyo desarrollo tienes poco tiempo. Descubres que tu sistema puede ejecutar el código necesario en el peor caso el doble de rápido de lo que se requiere como básico.

- 1) ¿Cuánta energía ahorras si ejecutas a la velocidad actual y apagas el sistema cuando la computación se haya terminado?
- 2) ¿Cuánta energía dinámica ahorras si fijas el voltaje y la frecuencia a la mitad del máximo?

Ejercicio 1 (solución)

- 1) ¿Cuánta energía ahorras si ejecutas a la velocidad actual y apagas el sistema cuando la computación se haya terminado?

Se reduce el tiempo a la mitad, así que 50%

- 1) ¿Cuánta energía dinámica ahorras si fijas el voltaje y la frecuencia a la mitad del máximo?

✓ Tema 1 -> Energía dinámica

- Paso de un transistor de 0 -> 1 o de 1 -> 0
- Proporcional a $\frac{1}{2} \times CC \times \text{Voltaje}^2$

Así que: $\text{Energía} = \text{Energía}_{new} / \text{Energía}_{old} =$
 $\frac{1}{2} \times CC \times (\text{Voltaje}/2)^2 / (\frac{1}{2} \times CC \times (\text{Voltaje})^2) = 0,25$

Ejercicio 2

Se dispone de un computador con un solo núcleo que ejecuta una aplicación de evaluación de simulación numérica. Esta aplicación es intensiva en cálculo, a lo que dedica el **90% del tiempo**. El 10% restante lo dedica a esperar en operaciones de entrada/salida a disco.

Del tiempo que la aplicación pasa ejecutando instrucciones de cálculo un **75%** del tiempo lo pasa ejecutando operaciones en **coma flotante** y un 25% lo pasa ejecutando otras instrucciones. La ejecución de **una instrucción de coma flotante requiere como promedio 12 CPI**. El resto de las instrucciones requieren como promedio **4 CPI**.

Se está valorando la migración de esta aplicación a la siguiente alternativa, que no incorpora ninguna mejora para el tiempo de las operaciones de entrada/salida a disco:

- Un procesador con un solo núcleo y con una **frecuencia de reloj un 50% más alta** que la de la máquina original en el que las instrucciones de **coma flotante** requieren un **10% más de ciclos por instrucción** y el resto de instrucciones requieren un **25% más de ciclos por instrucción**.
- ¿Cuál será la **aceleración/desaceleración global** de la aplicación al migrarla a este nuevo procesador?

Ejercicio 2 (Solución)

- Primero hay que calcular el tiempo consumido en la máquina original (T_{orig}) considerando que T es el periodo de la señal de reloj e I el número de instrucciones del programa:

$$T_{orig} = I \times 0,75 \times 12 \times T + I \times 0,25 \times 4 \times T = I \times (9 + 1) \times T$$

- El tiempo dedicado a ejecutar instrucciones en el nuevo computador será T_{new} :

$$T_{new} = ((0,75 \times 1,1 \times 12) + (0,25 \times 1,25 \times 4)) \times I \times T / 1,5 = 11,15 / 1,5 \times I \times T$$

$$T_{new} = 7,433 \times I \times T$$

- El *speedup* (aceleración) debido a instrucciones será S :

$$S = T_{orig} / T_{new} = 10 / 7,433 = 1,345$$

Tenemos que aplicar la ley de Amdahl que vimos en Fundamentos de Computadores para calcular el Speedup total.

$S_t = 1 / ((1 - F_m) + F_m / A_m)$ siendo F_m la fracción de programa mejorada y A_m la mejora aplicada a dicha fracción.

$$\text{En este caso: } S_t = 1 / (0,1 + 0,9 / 1,345) = 1,3$$

Ejercicio 3

Supongamos que se ejecutan operaciones en punto flotante (FP) con una frecuencia de 25% y para ellas el CPI medio es 4. El CPI medio para las otras instrucciones es de 1,33.

La frecuencia con que se ejecuta la instrucción FSQRT es del 2% sobre las operaciones en FP y su CPI es de 20.

Nos proponen dos alternativas para bajar el CPI. Una es bajar el CPI de FSQRT a 2 y otra bajar el CPI medio de todas las operaciones en FP a 2,5.

Comparar estas dos alternativas de diseño comparando tiempos de ejecución, es decir, usando la ecuación que evalúa el tiempo de CPU a partir de los valores de CPI vista en clase (tema 1).

Ejercicio 3 (solución)

Opción1.

Primero hay que observar que la frecuencia de reloj no cambia y tampoco el número de instrucciones, solo el CPI. Vamos a calcular el CPI original para todo el código:

$$CPI_{original} = ((4 \times 0,25) \times I + (1,33 \times 0,75) \times I) / I = 1,99$$

Podemos calcular el CPI para la versión con FQSRT restando el número de ciclos ahorrado del CPI original:

$$CPI_{op1} = CPI_{con_FSQRT_mejorado} =$$

$$CPI_{original} - 0,02 \times 0,25 \times (CPI_{old_FSQRT} - CPI_{new_FSQRT}) = 1,99 - 0,02 \times 0,25 \times (20 - 2) = 1,90$$

Opción2.

Ahora podemos calcular el CPI de la segunda opción que propone el enunciado, o sea, para la mejora de todas las instrucciones en punto flotante del mismo modo o sumando el CPI de las instrucciones FP y no FP. Usando esto último obtenemos:

$$CPI_{op2} = (2,50 \times 0,25) + (0,75 \times 1,33) = 1,62$$

Ejercicio 3 (solución)

Dado que el CPI de esta segunda opción es ligeramente más bajo, su rendimiento será marginalmente mejor. Más en concreto el speedup para esta segunda opción será:

$$\begin{aligned}\text{Speedup}_2 &= \text{CPUtime_original} / \text{CPUtime_newFP} = \\ &= (I \times \text{ciclo reloj} \times \text{CPI}_{\text{original}}) / (I \times \text{ciclo de reloj} \times \text{CPI}_{\text{new_FP}}) = \\ &= 1,99 / 1,62 = 1,23\end{aligned}$$

Este ejercicio se podría haber resuelto también usando la ley de Amdahl.

Ejercicio 4

- Un programa corre en un procesador segmentado presentando un CPI de 1.6. El sistema admite hasta una instrucción por ciclo. El programa ejecuta un 20% de instrucciones de salto y la tasa de acierto en la predicción es del 75%. El coste de un salto mal predicho es de 6 ciclos, y el coste de los saltos bien predichos es cero.
 - a) Calcula el CPI si mejoramos el predictor de saltos para que tenga una tasa de aciertos del 90%.
 - b) Calcular el CPI si además de lo anterior se aplica la **técnica de salto retardado** suponiendo que, en el código, el compilador siempre encuentra una instrucción para aplicarla, y consigue el máximo beneficio que ésta permite.

Ejercicio 4. Respuesta

- Clave: hay un sobrecoste asociado a otro tipo de paradas que hay que obtener de los datos del enunciado.
 - a) Con el sobrecoste calculado y la fórmula del CPI obtenemos
 $\text{CPI} = 1,42$
 - b) Hay que considerar a mayores que en este caso el coste de cada salto será un ciclo menos porque uno de ellos se aprovecha con el salto retardado con una ranura de retardo. Serán 5 ciclos por cada salto mal predicho. Esto nos da un
 $\text{CPI} = 1,4$