



Documento anónimo

## Examen de problemas Grupo A 2014 resuelto.pdf

*Exámenes Resueltos (teoría y Prácticas)*



2º Arquitectura de Computadores



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de  
Telecomunicación  
UGR - Universidad de Granada



## MÁSTER EN DATA SCIENCE

¿Quieres ser el **profesional más  
demandado** del siglo XXI?

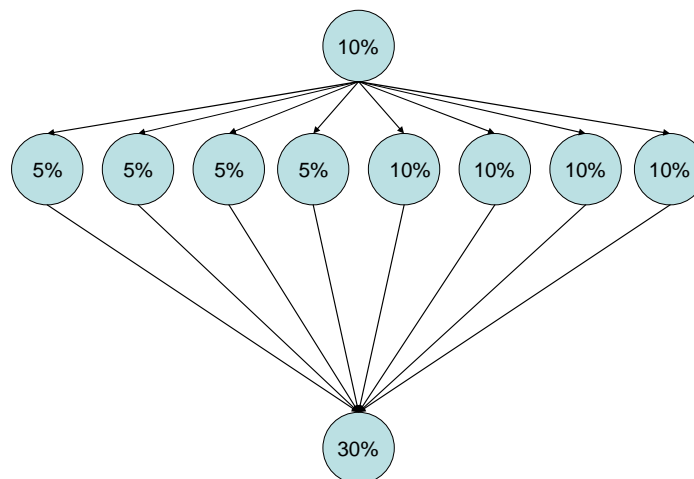
[www.cunef.edu](http://www.cunef.edu)

## ARQUITECTURA DE COMPUTADORES

Benchmarks 2014.

**Problema.** En la Figura se muestra el grafo de dependencia entre tareas para una aplicación, indicando en cada nodo la fracción del tiempo de ejecución secuencial que la aplicación tarda en ejecutar cada grupo de tareas del grafo. Suponiendo un tiempo de ejecución secuencial de 200 s, que las tareas no se pueden dividir en tareas de menor granularidad y un tiempo de comunicación despreciable:

- ¿Cuál es el valor de  $f$  en la ley de Amdahl?
- ¿Cuál es el número mínimo de procesadores para el que se obtiene la máxima ganancia de velocidad?
- ¿Cuál es esa ganancia máxima?



**Solución:**

$$f = (\text{parte del tiempo secuencial no paralelizable}) / (\text{tiempo secuencial}) = 200 * (0.1 + 0.3) / 200 = 0.4$$

Número mínimo para el que se consigue la máxima ganancia: **6 procesadores (cuatro procesadores ejecutarían las tareas paralelas del 10% y dos procesadores podrían ejecutar en el mismo tiempo las cuatro del 5%: si se meten más procesadores para hacer las tareas del 5%, el tiempo no se reduciría dado que tendría que esperar a que terminen los que ejecutan las tareas del 10%)**

$$S = T_1 / T_p = 200s / (200s * (0.1 + 0.1 + 0.3)) = 1 / 0.5 = 2$$

**Problema.** En un multiprocesador SMP con 4 procesadores o nodos (N1-N4) basado en un bus, que implementa el protocolo MESI para mantener la coherencia de cache, cada procesador dispone de una cache de datos de 512 Kbytes con marcos de bloque (también llamados líneas) de 32 bytes.

En el multiprocesador se están ejecutando en paralelo cuatro hebras, cada una en un procesador, que acceden a los elementos de dos arrays  $X[]$  e  $Y[]$  de 16 elementos de 32 bits, y se encuentran almacenados en posiciones consecutivas de la memoria principal (primero los de  $X[]$  y luego los de  $Y[]$ ) a partir de un inicio de bloque.

Indique los estados de este bloque en las caches ante la siguiente secuencia de eventos:

- Lectura generada por el procesador 1 a  $X[0]$
- Lectura generada por el procesador 2 a  $Y[2]$

- Escritura generada por el procesador 1 a X[4]
- Escritura generada por el procesador 2 a Y[0]
- Escritura generada por el procesador 3 a X[10]
- Lectura generada por el procesador 4 a Y[1]

NOTA: Suponga que bloques distintos se almacenan en la cache de cada procesador en marcos de bloque (líneas) diferentes.

**Solución:**

Cada bloque de cache tiene 32 Bytes, como cada dato tiene 4 Bytes (son datos de 32 bits), en un bloque de cache hay espacio para  $32/4=8$  datos. Por lo tanto los datos en memoria se ubicarían de la siguiente manera:

B(1)	X[0] .... X[7]
B(2)	X[0] .... X[15]
B(3)	Y[0] .... Y[7]
B(4)	Y[8] .... Y[15]

Los estados en la cache son los siguientes:

Acceso	Bloque	C1	C2	C3	C4
R1(X[0])	R1(B1)	(B1,E)			
R2(Y[2])	R2(B3)	(B1,E)	(B3,E)		
W1(X[4])	W1(B1)	(B1,M)	(B3,E)		
W2(Y[0])	W2(B3)	(B1,M)	(B3,M)		
W3(X[10])	W3(B2)	(B1,M)	(B3,M)	(B2,M)	
R4(Y[1])	R4(B3)	(B1,M)	(B3,S)	(B2,M)	(B3,S)

**Problema.** En un multiprocesador SMP con 4 procesadores basado en un bus, que implementa el protocolo MESI para mantener la coherencia, supongamos una dirección de memoria incluida en un bloque que no se encuentra en ninguna cache. Indique los estados de este bloque en las caches ante la siguiente secuencia de eventos para dicha dirección:

- Escritura generada por el procesador 1
- Lectura generada por el procesador 2
- Escritura generada por el procesador 1
- Escritura generada por el procesador 2
- Escritura generada por el procesador 3
- Lectura generada por el procesador 1

¿Por qué estados se pasaría si se utilizase el protocolo MSI?

Acceso	C1	C2	C3	C4
W1	M			
R2	S	S		
W1	M	I		
W2	I	M		
W3	I	I	M	
R1	S	I	S	

Para el MSI se pasa por los mismos estados: no se utiliza el estado E en el caso anterior para MESI

**Problema.** Suponga que un programa representativo del uso de un computador con un procesador no segmentado contiene un 20% de instrucciones de carga de memoria (LOAD) que consumen 5 ciclos; un 15% de escrituras en memoria (STORES) que consumen 4 ciclos; un 40% de operaciones en coma flotante que precisan 7 ciclos; un 15% de instrucciones de operaciones con enteros que necesitan 5 ciclos. El resto de instrucciones necesitan 4 ciclos. (a) ¿Qué dato o datos necesita para conocer el tiempo que tarda en ejecutarse el programa (en segundos)? (b) Indique el valor de f para la ley de Amdahl para la mejora de prestaciones en las instrucciones en coma flotante, y el valor de f para la mejora en las instrucciones de operaciones con enteros. (c) ¿Qué intentaría mejorar, las instrucciones en coma flotante o las de operaciones con enteros? ¿por qué?.

**Solución:**

- Como  $T_{cpu} = NI * CPI * T_{ciclo}$  hacen falta el número de instrucciones, NI, y el tiempo de ciclo,  $T_{ciclo}$  (o lo que es lo mismo, la frecuencia de reloj).
- Los valores de f para la ley de Amdahl se obtienen a partir de la fracción del tiempo secuencial en el que no se puede aplicar la mejora (o lo que es lo mismo, 1 menos la fracción en la que se puede aplicar la mejora)

$$f(FP) = 1 - ((NI * 0.4 * 7 * T_{ciclo}) / (NI * (0.2 * 5 + 0.15 * 4 + 0.4 * 7 + 0.15 * 5 + 0.1 * 4) * T_{ciclo})) \\ = (1 - (0.4 * 7 / (0.2 * 5 + 0.15 * 4 + 0.4 * 7 + 0.15 * 5 + 0.1 * 4)))$$

$$f(INT) = 1 - ((NI * 0.15 * 5 * T_{ciclo}) / (NI * (0.2 * 5 + 0.15 * 4 + 0.4 * 7 + 0.15 * 5 + 0.1 * 4) * T_{ciclo})) \\ = (1 - (0.15 * 5 / (0.2 * 5 + 0.15 * 4 + 0.4 * 7 + 0.15 * 5 + 0.1 * 4)))$$

- c) Como  $f(FP) < f(INT)$  es mejor **reducir el tiempo de las instrucciones en coma flotante** porque la parte del tiempo en el que no se podría aplicar esa mejora es menor.