Documento anónimo

Examen de problemas Grupo A 2013 resuelto.pdf

Exámenes Resueltos (teoría y Prácticas)

2° Arquitectura de Computadores

Grado en Ingeniería Informática

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

ARQUITECTURA DE COMPUTADORES (Grupos A y B. Ing. Informática)
Benchmark 2013

1. En un procesador no segmentado que funciona a 600 MHz, hay un 30% de instrucciones LOAD que necesitan 4 ciclos, un 5% de instrucciones STORE que necesitan 3 ciclos, un 30% de instrucciones con operaciones de enteros que necesitan 6 ciclos, un 30% de instrucciones con operandos en coma flotante que necesitan 8 ciclos, y un 5% de instrucciones de salto que necesitan 3 ciclos. Las operaciones con enteros se realizan en una ALU con un retardo de 4 ciclos, y las de coma flotante en una unidad funcional con un retardo de 6 ciclos.
(a) ¿Qué es mejor, reducir un 25% el retardo de la ALU para enteros o un 50% el retardo de la unidad funcional de coma flotante?. (b) ¿Cuál es la máxima ganancia que se puede obtener por reducción en el tiempo de las operaciones con enteros en la ALU?. (c) ¿Y por reducción en el tiempo de las operaciones en la unidad de coma flotante?.

SOLUCIÓN:
(a) El tiempo de CPU de la situación inicial es:

$$T_{CPU}=NI*(0.3*4+0.05*3+0.3*6+0.3*8+0.05*3)*T_{ciclo}$$

En el caso de reducir el tiempo de la ALU un 25%, como necesita 4 ciclos, pasaría a 4*0.75=3 ciclos, que sumados al resto de ciclos (6-4=2) de la instrucción que no se ven afectados por la mejora, harían que el nuevo CPI de las instrucciones de operación con la ALU sea 5, y

$$T_{CPU\_ALU}=NI*(0.3*4+0.05*3+0.3*5+0.3*8+0.05*3)*T_{ciclo}$$

Cuando se reduce el tiempo de la unidad de coma flotante (FP) un 50%, como necesita 6 ciclos, pasaría a 6*0.50=3 ciclos, que sumados al resto de ciclos (8-6=2) de la instrucción que no se ven afectados por la mejora, harían que el nuevo CPI de las instrucciones de operación en coma flotente sea 5, y

$$T_{CPU\_FP}=NI*(0.3*4+0.05*3+0.3*6+0.3*5+0.05*3)*T_{ciclo}$$

Por tanto, como $T_{CPU\_FP} < T_{CPU\_ALU}$ es mejor la reducción del retardo de la unidad de coma flotante.

(b) La mayor ganancia de velocidad al reducir el tiempo de retardo de la ALU se conseguiría cuando dicho tiempo sea igual a 0. Por lo tanto, las instrucciones de operación con la ALU stendrían un CPI=2 ciclos, por tanto:

$$S_{ALU\_max}= (0.3*4+0.05*3+0.3*6+0.3*8+0.05*3)/(0.3*4+0.05*3+0.3*\mathbf{2}+0.3*8+0.05*3)=1.27$$

(c) La mayor ganancia de velocidad al reducir el tiempo de retardo de la unidad funcional de coma flotante se conseguiría cuando dicho tiempo sea igual a 0. Por lo tanto, las instrucciones de operación en coma flotante tendrían un CPI=2 ciclos, y se tendría:

$$S_{FP\_max}= (0.3*4+0.05*3+0.3*6+0.3*8+0.05*3)/(0.3*4+0.05*3+0.3*6+0.3*\mathbf{2}+0.05*3)=1.46$$

Como se puede ver, es mayor la ganancia máxima que se consigue al reducir el retardo de la unidade funcional de coma flotante que el retardo de la ALU.

2. En un multiprocesador CC-NUMA con 4 procesadores o nodos (N1-N4) que implementa el protocolo MSI basado en directorios, sin difusión, 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 que acceden a los elementos de dos arrays X[] e Y[] de 16 elementos de 32 bits, que se encuentran almacenados en posiciones consecutivas de la memoria principal del nodo N2 (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, y los cambios en los contenidos del directorio ante la siguiente secuencia de eventos:
- Lectura generada por el procesador 1 a X[1]
- Lectura generada por el procesador 2 a X[2]
- Escritura generada por el procesador 1 a X[7]
- Escritura generada por el procesador 2 a Y[0]
- Escritura generada por el procesador 3 a X[14]
- Lectura generada por el procesador 4 a Y[4]

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

SOLUCIÓN:

Teniendo en cuenta que los datos son de 32 bits, es decir, 4 bytes, en cada bloque (32 bytes) se encuentran 8 datos (32 bytes por bloque / 4 bytes por dato)

Por lo tanto tendríamos los datos de los dos arrays ubicados en cuatro bloques que denominamos B1, B2, B3, y B4, según la figura:

X[0] … X[7]	B1
X[8] … X[15]	B2
Y[0] … Y[7]	B3
Y[8] … Y[15]	B4

Los accesos que se realizan son:

(1) R1(B1) El procesador 1 lee X[1] que está en el Bloque B1
(2) R2(B1) El procesador 2 lee X[2] que está en el Bloque B1

(3) W1(B1) El procesador 1 escribe en X[7] que está en el Bloque B1
(4) W2(B3) El procesador 2 escribe en Y[0] que está en el Bloque B3
(5) W3(B2) El procesador 3 escribe en X[14] que está en el Bloque B2
(6) R4(B1) El procesador 4 lee Y[4] que está en el Bloque B

Los estados en los que quedan los bloques en cache tras cada acceso se muestran en la siguiente Tabla. Hay que tener en cuenta que sólo los accesos que se producen al mismo bloque son los que pueden dar problemas de coherencia (por tanto en protocolo se aplica de forma independiente para los accesos a cada bloque):

Acceso	Cache P1	Cache P2	Cache P3	Cache P4
(1) R1(B1)	B1: S			
(2) R2(B1)	B1: S	B1: S		
(3) W1(B1)	B1: M	B1: I		
(4) W2(B3)	B1:M	B1: I B3: M		
(5) W3 (B2)	B1: M	B1: I B3: M	B2: M	
(6) R4(B3)	B1: M	B1: I B3: S	B2: M	B3: S

Como se puede ver, dado un bloque Bi, solo puede estar en estado M en una cache.

En cuanto al estado del **directorio que se encuentra en el nodo N2** para controlar la ubicación de los Bloques y su estado de validez, la evolución es la siguiente:

Inicialmente, todos los bloques (B1, B2, B3, B4) son válidos en la memoria local de N2 y no están en ninguna cache. Luego evolucionan según el acceso (teniendo en cuenta en los bloques es los que hay copia y se esta se encuentra en estado S, y entonces debe haber copia válida en la memoria local, o en estado M, y en ese caso la copia en memoria local no es válida).

Inicialmente

	N1	N2	N3	N4	Val
B1	0	0	0	0	1
B2	0	0	0	0	1
B3	0	0	0	0	1
B4	0	0	0	0	1

(1) R1(B1)

	N1	N2	N3	N4	Val
B1	1	0	0	0	1
B2	0	0	0	0	1
B3	0	0	0	0	1
B4	0	0	0	0	1

(2) R2(B1)

	N1	N2	N3	N4	Val
B1	1	1	0	0	1
B2	0	0	0	0	1
B3	0	0	0	0	1
B4	0	0	0	0	1

(3) W1(B1)

	N1	N2	N3	N4	Val
B1	1	0	0	0	0
B2	0	0	0	0	1
B3	0	0	0	0	1
B4	0	0	0	0	1

(4) W2(B3)

	N1	N2	N3	N4	Val
B1	1	0	0	0	0
B2	0	0	0	0	1
B3	0	1	0	0	0
B4	0	0	0	0	1

(5) W3(B2)

	N1	N2	N3	N4	Val
B1	1	0	0	0	0
B2	0	0	1	0	0
B3	0	1	0	0	0
B4	0	0	0	0	1

(6) R4(B3)

	N1	N2	N3	N4	Val
B1	1	1	0	0	0
B2	0	0	1	0	0
B3	0	1	0	1	1
B4	0	0	0	0	1