



Documento anónimo

AC - Examen parcial 2012 - Mancia.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

2º curso / 2º cuatr.

Grado en
Ing. Informática

Arquitectura de Computadores: Exámenes y Controles

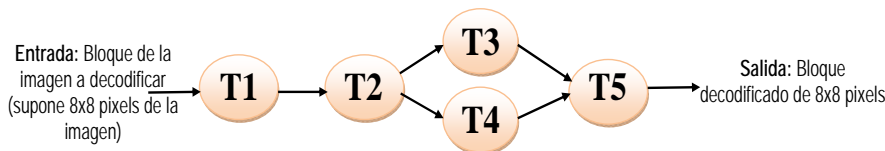
Control Tems 1 a 3 (grupos D y E) del 01/06/2012 resuelto

Material elaborado por los profesores responsables de la asignatura:
Mancia Anguita, Julio Ortega

Licencia Creative Commons 

1 Enunciado Control Tems 1 a 3 (grupos D y E) del 01/06/2012

Ejercicio 1. Se va a paralelizar un decodificador JPEG en un multiprocesador. Se ha extraído para la aplicación el siguiente grafo de tareas que presenta una estructura segmentada (o de flujo de datos):



La tareas 1, 2, 3 y 5 se ejecutan en un tiempo igual a t , mientras que la tarea 4 supone $1,5t$

El decodificador JPEG aplica el grafo de tareas de la figura a bloques de la imagen, cada uno de 8×8 píxeles. Si se procesa una imagen que se puede dividir en n bloques de 8×8 píxeles, a cada uno de esos n bloques se aplica el grafo de tareas de la figura. Obtenga la mayor ganancia en prestaciones que se puede conseguir paralelizando el decodificador JPEG en (suponga despreciable el tiempo de comunicación/sincronización): **(a)** 5 procesadores, **(b)** 4 procesadores y **(c)** 3 procesadores. En los tres casos la ganancia se tiene que calcular suponiendo que se procesa una imagen con un total de n bloques de 8×8 píxeles.

Ejercicio 2. Suponga que en un CC-NUMA de red estática de 4 nodos (N0-N3) se implementa un protocolo MSI basado en directorio distribuido sin difusión con dos estados en el directorio (válido e inválido). Cada nodo tiene 4 GBytes de memoria y una línea de cache supone 64 Bytes. Calcule el tamaño del subdirectorio de uno nodo en bytes suponiendo que el directorio utiliza vector de bits completo.

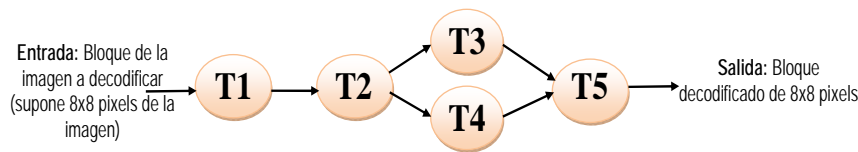
Cuestión 1. Indique cuál es la diferencia fundamental entre una arquitectura CC-NUMA y una arquitectura SMP.

Cuestión 2. Suponga un multiprocesador con el protocolo MESI de espionaje. Si el procesador de un nodo escribe en un bloque que tiene en su cache en estado I, debe (indique cuál sería la respuesta correcta y razone por qué es la respuesta correcta):

- Generar paquete de petición de acceso E al bloque y pasar el bloque a M
- Generar paquete de petición de acceso E y pasar el bloque a estado E
- Generar paquete de petición de acceso E con lectura y pasar el bloque a estado E
- Generar paquete de petición de acceso E al bloque con lectura y pasar el bloque a M

2 Solución Control Temás 1 a 3 (grupos D y E) del 01/06/2012

Ejercicio 1. Se va a paralelizar un decodificador JPEG en un multiprocesador. Se ha extraído para la aplicación el siguiente grafo de tareas que presenta una estructura segmentada (o de flujo de datos):



Las tareas 1, 2, 3 y 5 se ejecutan en un tiempo igual a t , mientras que la tarea 4 supone $1,5t$

El decodificador JPEG aplica el grafo de tareas de la figura a bloques de la imagen, cada uno de 8×8 píxeles. Si se procesa una imagen que se puede dividir en n bloques de 8×8 píxeles, a cada uno de esos n bloques se aplica el grafo de tareas de la figura. Obtenga la mayor ganancia en prestaciones que se puede conseguir paralelizando el decodificador JPEG en (suponga despreciable el tiempo de comunicación/sincronización): **(a)** 5 procesadores, **(b)** 4 procesadores y **(c)** 3 procesadores. En los tres casos la ganancia se tiene que calcular suponiendo que se procesa una imagen con un total de n bloques de 8×8 píxeles.

Solución

Para obtener la ganancia se tiene que calcular el tiempo de ejecución en secuencial para un tamaño del problema de n bloques $T_s(n)$ y el tiempo de ejecución en paralelo para un tamaño del problema de n y los p procesadores indicados en (a), (b) y (c) $T_p(p, n)$.

Tiempo de ejecución secuencial $T_s(n)$. Un procesador tiene que ejecutar las 5 tareas para cada bloque de 8×8 píxeles. El tiempo que dedica el procesador a cada bloque es de $5,5t = 4t$ (tareas 1, 2, 3 y 5) + $1,5t$ (tarea 4), luego para n bloques el tiempo de ejecución secuencial será el siguiente:

$$T_s(n) = n \times (5,5t)$$

(a) Tiempo de ejecución paralelo y ganancia en prestaciones para 5 procesadores $T_p(5, n)$, $S(5, n)$. Cada tarea se asigna a un procesador distinto, por tanto todas se pueden ejecutar en paralelo (ver asignación de tareas a procesadores en Tabla 1). El tiempo de ejecución en paralelo en un cauce consta de dos tiempos: el tiempo que tarda en procesarse la primera entrada (ver celdas con fondo verde en la tabla) + el tiempo que

Tabla 1. Asignación de tareas a procesadores, ocupación de las etapas del cauce para los primeros bloques procesados y tiempo de procesamiento del primer bloque y de los siguientes para 5 procesadores

<p>Asignación de tareas a 5 procesadores</p>					
Etapas 1 (t)	Etapas 2 (t)	Etapas 3 (1,5t=max{1t,1,5t})	Etapas 4 (t)	Terminado	Tiempo proces.
T1 (procesador P1)	T2 (procesador P2)	T3 (procesador P3) y T4 (procesador P4)	T5 (procesador P5)		
0 - Bloque 1 - t					Procesamiento Bloque 1 = latencia del cauce = 4,5t
t - Bloque 2 - 2t	t - Bloque 1 - 2t				
2t - Bloque 3 - 3t	2t - Bloque 2 - 3t	2t - Bloque 1 - 3,5t			
3t - Bloque 4 - 4t	3t - Bloque 3 - 4t	3,5t - Bloque 2 - 5t	3,5t - Bloque 1 - 4,5t		
4t - Bloque 5 - 5t	4t - Bloque 4 - 5t	5t - Bloque 3 - 6,5t	5t - Bloque 2 - 6t	Bloque 1	$t + t + 1,5t + t = 4,5t$
5t - Bloque 6 - 6t	5t - Bloque 5 - 6t	6,5t - Bloque 4 - 8t	6,5t - Bloque 3 - 7,5t	Bloque 2	$4,5t + 1,5t = 6,0t$
6t - Bloque 7 - 7t	6t - Bloque 6 - 7t	8t - Bloque 5 - 9,5t	8t - Bloque 4 - 9t	Bloque 3	$6t + 1,5t = 7,5t$
...	Bloque 4	$7,5t + 1,5t = 9,0t$
T_entrada_etapa - Bloque x - T_salida_etapa (Ej. t - Bloque 1 - 2t)					

tardan el resto de entradas al cauce (bloques de 8x8 píxeles en esta aplicación) en ir terminando una detrás de otra. Este último depende de la etapa más lenta, que en este caso es la Etapa 3, que supone $1,5t$ ($=\text{máximo}\{\text{tiempo } T3, \text{ tiempo } T4\}$). La Etapa 3 consta de 2 tareas que se ejecutan en paralelo ($T3$ y $T4$). La tarea 5 empezará su ejecución, según el grafo, cuando acaben las tareas 3 y 4; por tanto, el tiempo de la Etapa 3 del cauce depende del tiempo que tarda la tarea $T4$ que es la más lenta de las dos que incluye. El tiempo de procesamiento y la ganancia serían:

$$T_p(5, n) = 4,5t + (n - 1) \times 1,5t = 3t + n \times 1,5t$$

$$S(5, n) = \frac{T_s(n)}{T_p(5, n)} = \frac{n \times 5,5t}{3t + n \times 1,5t} \xrightarrow{n \rightarrow \infty} 3,6 \text{ (para } n \text{ suficientemente grande)}$$

(b) Tiempo de ejecución paralelo y ganancia en prestaciones para 4 procesadores $T_p(4, n)$, $S(4, n)$. Se deben asignar las 5 tareas a los 4 procesadores de forma que se consiga el mejor tiempo de ejecución paralelo, es decir, el menor tiempo por bloque una vez procesado el primero. Se puede conseguir el menor tiempo por bloque, por ejemplo, asignando $T1$ y $T2$ a un procesador (ver asignación de tareas a procesadores en la Tabla 2). Con esta asignación se unen la Etapa 1 y la Etapa 2 del cauce del caso (a) en una única etapa. El tiempo paralelo y la ganancia conseguidas son:

$$T_p(4, n) = 4,5t + (n - 1) \times 2t = 2,5t + n \times 2t$$

$$S(4, n) = \frac{T_s(n)}{T_p(4, n)} = \frac{n \times 5,5t}{2,5t + n \times 2t} \xrightarrow{n \rightarrow \infty} 2,75 \text{ (para } n \text{ suficientemente grande)}$$

Tabla 2. Asignación de tareas a procesadores, ocupación de las etapas del cauce para los primeros bloques procesados y tiempo de procesamiento del primer bloque y de los siguientes para 4 procesadores

<p>Possible asignación de tareas a 4 procesadores</p>				
Etapa 1 (2t)	Etapa 2 (1,5t)	Etapa 3 (t)	Terminado	Tiempo proces.
T1 y T2 (procesador P1)	T3 (procesador P2) y T4 (procesador P3)	T5 (procesador P4)		
0 - Bloque 1 - 2t				Procesamiento Bloque 1 = latencia del cauce = 4,5t
2t - Bloque 2 - 4t	2t - Bloque 1 - 3,5t			
4t - Bloque 3 - 6t	4t - Bloque 2 - 5,5t	3,5t - Bloque 1 - 4,5t		
6t - Bloque 4 - 8t	6t - Bloque 3 - 7,5t	5,5t - Bloque 2 - 6,5t	Bloque 1	$2t + 1,5t + t = 4,5t$
8t - Bloque 5 - 10t	8t - Bloque 4 - 9,5t	7,5t - Bloque 3 - 8,5t	Bloque 2	$4,5t + 2t = 6,5t$
10t - Bloque 6 - 12t	10t - Bloque 5 - 11,5t	9,5t - Bloque 4 - 10,5t	Bloque 3	$6,5t + 2t = 8,5t$
12t - Bloque 7 - 14t	12t - Bloque 6 - 13,5t	11,5t - Bloque 5 - 12,5t	Bloque 4	$8,5t + 2t = 10,5t$
...	Bloque 5	$10,5t + 2t = 12,5t$
$T_{\text{entrada_etapa}} - \text{Bloque } x - T_{\text{salida_etapa}}$				

(c) Tiempo de ejecución paralelo y ganancia en prestaciones para 3 procesadores $T_p(3, n)$, $S(3, n)$. Se deben asignar las 5 tareas a los 3 procesadores de forma que se consiga el mejor tiempo de ejecución paralelo. Se puede conseguir la mejor productividad (es decir, el menor tiempo por bloque), por ejemplo, asignando $T1$ y $T2$ a $P1$, $T4$ y $T5$ a $P2$ y $T3$ a $P3$ (ver asignación de tareas a procesadores en la Tabla 3). Con esta asignación hay dos etapas en el cauce: la Etapa 1 de $2t$ y la Etapa 2 de $2,5t$ ($=\text{máximo}\{t \text{ de } T3, 1,5t \text{ de } T4\} + t$). Obsérvese que la Etapa 2 supondría también $2,5t$ si se asignaran $T3$ y $T5$ a $P2$ y $T4$ a $P3$. Ocurre



esto porque T5 no puede empezar hasta que no hayan terminado T3 y T4 y como T4 es más lenta, la Etapa 2 del cauce va a suponer $2,5t = 1,5t$ de T4 + t de T5. El tiempo de ejecución paralelo y la ganancia son en este caso:

$$T_p(3, n) = 4,5t + (n - 1) \times 2,5t = 2t + n \times 2,5t$$

$$S(3, n) = \frac{T_s(n)}{T_p(3, n)} = \frac{n \times 5,5t}{2t + n \times 2,5t} \xrightarrow{n \rightarrow \infty} 2,2 \text{ (para } n \text{ suficientemente grande)}$$

Tabla 3. Asignación de tareas a procesadores, ocupación de las etapas del cauce para los primeros bloques procesados y tiempo de procesamiento del primer bloque y de los siguientes para 3 procesadores

<p>Possible assignment of tasks to 3 processors</p>			
Etapa 1 (2t)	Etapa 2 (2'5t = máximo{t, 1'5t}+t)	Terminado	Tiempo proces.
T1 y T2 (procesador P1)	T4 y T5 (procesador P2) y T3 (procesador P3) o T3 y T5 (procesador P2) y T4 (procesador P3)		
0 - Bloque 1 - 2t			Procesamiento Bloque 1 = latencia del cauce = 4,5t
2t - Bloque 2 - 4t	2t - Bloque 1 - 4,5t		
4t - Bloque 3 - 6t	4,5t - Bloque 2 - 7t		
6t - Bloque 4 - 8t	7t - Bloque 3 - 9,5t	Bloque 1	2t + 2,5t = 4,5t
8t - Bloque 5 - 10t	9,5t - Bloque 4 - 12t	Bloque 2	4,5t + 2,5t = 7,0t
10t - Bloque 6 - 12t	12t - Bloque 5 - 14,5t	Bloque 3	7t + 2,5t = 9,5t
12t - Bloque 7 - 14t	14,5t - Bloque 6 - 17t	Bloque 4	9,5t + 2,5t = 12,0t
...	...	Bloque 5	12t + 2,5t = 14,5t
T_entrada_etapa - Bloque x - T_salida_etapa			

Ejercicio 2. Suponga que en un CC-NUMA de red estática de 4 nodos (N0-N3) se implementa un protocolo MSI basado en directorio distribuido sin difusión con dos estados en el directorio (válido e inválido). Cada nodo tiene 4 GBytes de memoria y una línea de cache supone 64 Bytes. Calcule el tamaño del subdirectorio de uno nodo en bytes suponiendo que el directorio utiliza vector de bits completo.

Solución

Datos del ejercicio

- Tamaño Memoria principal por Nodo: 4 GB = 2^{32} B = TMN
- Tamaño Línea de Cache (bloque de memoria): 64 B = 2^6 B = TLC
- 1 bits de estado por bloque en el subdirectorio ya que hay que codificar dos estados (válido, inválido).
- 4 nodos => en el subdirectorio habrá un vector de 4 bits para cada bloque de memoria.

Cada entrada del subdirectorio tendrá 5 bits: 1 bit de estado + 4 bits del vector de bits completo (un bit asociado a cada nodo). El subdirectorio tendrá una entrada por cada bloque de memoria principal del nodo. Por tanto, cada subdirectorio tendrá TMN/TLC entradas de 5 bits cada una.

$$\text{Subdirect.}_\text{. nodo} = (1b + 4b) \times \frac{\text{TMN}}{\text{TLC}} = 5b \times \frac{2^{32} B}{2^6 B} = 5b \times 2^{26} = \frac{2^6 \times 5b}{2^3 (b/B)} \times 2^{20} = (2^3 \times 5) MB = 40 MB$$



Cuestión 1. Indique cuál es la diferencia fundamental entre una arquitectura CC-NUMA y una arquitectura SMP.

Solución.

Ambos, SMP y CC-NUMA, son multiprocesadores; es decir, tanto en un SMP como en un CC-NUMA los procesadores comparten todo el espacio de direcciones físico. También en ambos se mantiene coherencia entre caches de distintos procesadores (CC- *Cache-Coherence*).

En un SMP (*Symmetric Multiprocessor*) la memoria se encuentra centralizada en el sistema a igual distancia (latencia) de todos los procesadores (nodos) mientras que, en un CC-NUMA (*Cache-Coherence Non Uniform Memory Access*), cada procesador tiene físicamente cerca un conjunto de sus direcciones de memoria porque los módulos de memoria están físicamente distribuidos entre los procesadores (nodos) y, por tanto, todos los módulos no están a igual distancia (latencia) de todos los procesadores.

La memoria centralizada del SMP hace que el tiempo de acceso a una dirección de memoria en un SMP sea igual para todos los procesadores y que el tiempo de acceso a memoria de un procesador sea el mismo independientemente de la dirección a la que esté accediendo; por estos motivos se denomina multiprocesador simétrico (*Symmetric Multiprocessor*) o UMA (*Uniform Memory Access*).

Los módulos de memoria físicamente distribuidos entre los procesadores de un CC-NUMA hacen que el tiempo de acceso a memoria dependa de si el procesador accede a una dirección de memoria que está en la memoria principal física que se encuentra en el nodo de dicho procesador (cercana, por tanto, al procesador que accede) o en la memoria física de otro nodo. Por este motivo el acceso no es uniforme o simétrico y recibe el nombre de NUMA (*Non Uniform Memory Access*).



Cuestión 2. Suponga un multiprocesador con el protocolo MESI de espionaje. Si el procesador de un nodo escribe en un bloque que tiene en su cache en estado I, debe (indique cuál sería la respuesta correcta y razone por qué es la respuesta correcta):

- e) Generar paquete de petición de acceso E al bloque y pasar el bloque a M
- f) Generar paquete de petición de acceso E y pasar el bloque a estado E
- g) Generar paquete de petición de acceso E con lectura y pasar el bloque a estado E
- h) Generar paquete de petición de acceso E al bloque con lectura y pasar el bloque a M

Solución

(d) Generar paquete de petición de acceso E (exclusivo) al bloque con lectura y pasar el bloque a M (modificado)

Razonamiento:

(1) Paquete con lectura: como la cache del nodo no tiene copia válida del bloque (debido al estado inválido), el controlador de cache del nodo tiene que generar un paquete que incluya lectura.

(2) Paquete con acceso exclusivo: como se va a escribir en la copia que se reciba del bloque, el controlador de cache del nodo tiene que pedir acceso exclusivo para que las copias del bloque en otras caches sean invalidadas (MESI utiliza invalidación para propagar lo que se escribe en la copia de un bloque en una cache al resto de caches con copia del mismo). Esto es necesario porque las siguientes lecturas del bloque que se realicen en otros nodos deberán acceder a la última modificación del mismo (que es la que se va a realizar en este nodo) en lugar de acceder a copias no actualizadas del bloque que estén en sus respectivas caches.

(3) Pasar a estado modificado: como el nodo va a escribir en el bloque cuando lo reciba, será la única copia válida del bloque en el sistema, por tanto, deberá estar en estado modificado para que el controlador de cache sepa que debe responder con el bloque si se recibe alguna petición que incluya lectura del bloque.

