

Arquitectura e Ingeniería de los Computadores (Segundo cuatrimestre)

Examen de Junio (11/6/02) Teoría (Soluciones)

1. Calcular el aumento del rendimiento (*speed-up*) de un programa ejecutándose en una máquina con n procesadores, suponiendo que este programa tiene un 15% de parte serie (grado de paralelismo 1), un 25% con grado de paralelismo 4, un 20% con grado de paralelismo 6 y el resto (40%) es completamente paralelizable.

Las partes que tienen un grado de paralelismo mayor que 1 (serie) mejoran en proporción a este grado de paralelismo. Además se incluye una parte que es completamente paralelizable y que la podemos dividir en tantas partes como procesadores se pongan. Normalizando el tiempo de ejecución con un solo procesador a uno ($T(1)=1$), entonces se tienen las siguientes situaciones:

$$\begin{array}{lll} 1 < n < 5 & \text{---->} & T(n) = 0,15 + (0,25+0,20+0,40)/n & S(n)=T(1)/T(n) \\ 4 < n < 7 & \text{---->} & T(n) = 0,15 + 0,25/4 + (0,20+0,40)/n & S(n)=T(1)/T(n) \\ 6 < n & \text{---->} & T(n) = 0,15 + 0,25/4 + 0,20/6 + 0,40/n & S(n)=T(1)/T(n) \end{array}$$

2. Explicar la ventaja del modelo de consistencia de liberación frente a la consistencia débil en la que no se distingue entre las operaciones de sincronización de adquisición y liberación. Explica también en qué casos es más ventajoso utilizar la consistencia de procesador frente a la secuencial.

La consistencia de liberación distingue entre la sincronización de adquisición y la de liberación, de manera que se pueden paralelizar ciertas transacciones que en la débil deben venir en orden. Es aconsejable mirar lo explicado en clase para ver qué transacciones se benefician de esto.

Con la secuencial y de procesador ocurre algo parecido. La consistencia secuencial exige un orden global para todas las instrucciones que se lancen, en cambio la de procesador sólo exige un orden global para las instrucciones lanzadas por un procesador. Esto quiere decir que las transacciones lanzadas por diferentes procesadores pueden darse en paralelo aumentando el rendimiento. Mirar la teoría para ampliar.

3. Explica qué es una red de interconexión multietapa (MIN). Dibuja el esquema de las partes de las que se compone. Dibuja una red multietapa Omega con conmutadores de dos entradas y dos salidas, la red tiene 8 entradas y 8 salidas. (En la red Omega todas las etapas de permutación son de barajado perfecto salvo la última que no permuta nada.)

Esto está tal cual en los apuntes de teoría y lo explicado en clase.

4. ¿Por qué en los protocolos de actualización en general, las transferencias de actualización son de palabras sueltas en lugar de líneas completas de caché?
¿Por qué no se hace lo mismo en los protocolos de invalidación y se invalidan palabras sueltas en lugar de líneas completas

Los de actualización sólo actualizan palabras porque es lo único que se necesita actualizar y porque es muy sencillo de implementar en el bus (no requiere un hardware específico, etc). Además es mucho más rápido que actualizar líneas enteras (no se carga el bus de forma innecesaria).

En los de invalidación, en cambio, es preciso operar con líneas enteras puesto que si utilizásemos palabras necesitaríamos al menos los bits de estado para cada una de las palabras, con lo que la caché aumentaría mucho su tamaño y se complicaría también el control de la misma. Habría también conflictos en el caso de actualizar la memoria con líneas cuyas palabras pueden estar inválidas, modificadas o compartidas a la vez.