

INTRODUCCIÓN A LA CONSISTENCIA

Carlos Llamas Jaén

Introducción

- La complejidad de las operaciones de memorias crecen a medida que se complica el modelo de computador.
- En un procesador tradicional no hay apenas problemas al ejecutarse las instrucciones de forma secuencial.
- En procesadores superescalares se complica un poco más, ya que hay que tener en cuenta que diferentes operaciones no choquen entre ellas en el acceso o escritura a memoria.
- Finalmente dónde más se complica es en equipos multiprocesadores, donde los datos están replicados entre procesadores y estos a su vez se encuentran a una distancia distinta de memoria, lo que puede provocar que los datos realmente no sean los que el procesador esté viendo en un determinado momento.

Consistencia de memoria

- Para solucionar la problemática anterior surgen los distintos modelos de consistencia de memoria:
 - Reglas que especifican como son manipuladas las operaciones de memorias en un sistema multiprocesador.
 - Contrato entre el programador y el sistema. Los programadores lo utilizan para saber como va a responder el sistema y el sistema debe diseñarse para conformar estas reglas.

Tipos

- Existen varios tipos de consistencia:
 - Consistencia estricta
 - Consistencia secuencial
 - Consistencia causal
 - PRAM
 - Ordenamiento parcial de almacenamiento
 - Consistencia de caché (Coherencia)

Coherencia vs Consistencia

- La coherencia hace referencia a que un valor o bloque de datos es procesado por varios procesadores de forma que todos los procesadores son capaces de observar los valores correctos tras realizar modificaciones a ellos.
- Sin embargo la consistencia hace referencia a que todos y cada uno de los bits que forman los datos en un sistema son observables de forma correcta por todos los procesadores tras realizar modificaciones a ellos.
- En el caso de sistemas secuencialmente consistentes se da el caso de que si una localización en memoria es coherente no tiene porque ser consistente, pero al revés sí.

Ejemplo

- Veamos ahora un ejemplo de esta última afirmación.
- Tenemos un código que trabaja sobre dos variables en dos procesadores distintos. En el primer procesador se leen los datos, en el segundo se guardan. Inicialmente VAR_A y VAR_B son iguales 5.
- P1:
 - CARGA VAR_A (Devuelve 5)
 - CARGA VAR_B (Devuelve 11)
- P2:
 - VAR_A = 10
 - VAR_B = 11
- Supongamos que el sistema es coherente, es decir si P2 modifica VAR_A, P1 debe ver 10 y no 5. Entonces deduciendo de lo anterior obtenemos la siguiente traza.

Ejemplo

- Se ha ejecutado en el siguiente orden:

1. Carga VAR_A
2. VAR_A = 10
3. VAR_B = 11
4. Carga VAR_B

Este código es coherente, pero no es consistente secuencialmente, debido a que para que lo sea, el orden correcto sería que se ejecutase cada procesador secuencialmente.

Ejemplo

- Cualquiera de los órdenes correctos serían los siguientes:

1. VAR_A = 10
2. VAR_B = 11
3. Carga VAR_A (Devuelve 10)
4. Carga VAR_B (Devuelve 11)

O viceversa

1. Carga VAR_A (Devuelve 5)
2. Carga VAR_B (Devuelve 5)
3. VAR_A = 10
4. VAR_B = 11

Referencias

- <http://ocw.uc3m.es/ingenieria-informatica/arquitectura-de-computadores/materiales/es-m5-02-consist-ocw.pdf>
- http://www.exa.unicen.edu.ar/catedras/arqui2/arqui2/filminas/Sistemas%2ode%2oProcesadores%2oparalelos_PARTE%2oll.pdf
- https://www.fdi.ucm.es/profesor/rhermida/AC-Grado/AC_tema6.pdf
- https://es.wikipedia.org/wiki/Coherencia_de_cach%C3%A9
- https://en.wikipedia.org/wiki/Consistency_model
- <https://cs.stackexchange.com/questions/20044/memory-consistency-vs-cache-coherence>