

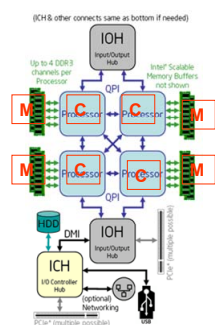
ARQUITECTURA DE COMPUTADORES BENCHMARK del TEMA 3

Un microprocesador multinúcleo no incluye memoria cache

(F)

El servidor de la figura tiene una arquitectura de tipo UMA

(F)



Señale en la figura (con una C) dónde está la memoria cache

Señale en la figura (con un M) dónde está la memoria principal

En un microprocesador SMT (Multihebra simultánea), en un instante determinado se pueden enviar a ejecutar instrucciones de hebras diferentes

(V)

En un microprocesador SMT (multihebra simultánea), se procesan varias hebras concurrentemente aunque en un instante determinado solo se pueden enviar a ejecutar instrucciones de la misma hebra.

(F)

En el protocolo MESI para mantener la coherencia de cache una línea puede estar en el estado S solo en una de las caches del multiprocesador

(F)

En el protocolo MESI para mantener la coherencia de cache, una línea puede estar en el estado E en varias caches del multiprocesador

(F)

En el protocolo MSI, si en la cache de un nodo N1 hay un bloque en estado M (Modificado), y ese nodo detecta que otro procesador intenta leer un dato que está en ese bloque, el bloque pasa al estado I (Inválido) en el nodo N1

(F)

En un multiprocesador NUMA con 16 nodos, 4 GBytes por nodo, y líneas de cache de 64 Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

$$2^2 \cdot 2^{30} / 2^6 = 2^{26} \text{ entradas}$$

En el multiprocesador NUMA descrito en la pregunta anterior ¿Cuántos bits tiene cada una de las entradas del directorio que se utiliza para mantener la coherencia de cache?

$$16 + 1 = 17$$

¿Qué valores se puede observar en R si el modelo de consistencia de memoria del computador donde están los procesadores que ejecutan estos códigos no respeta el orden $W \rightarrow W$ (sí respeta los demás), e inicialmente $X=Y=0$?

P1:	X=2	P2:	R=1;
	Y=1		if (Y==1) R=X;

R=0; R=1; R=2

Si respeta el orden $W \rightarrow W$: R=1; R=2

¿Qué valor deben r1, r2, y r3 para que la secuencia de instrucciones siguiente implemente un cerrojo (lock(k)) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto.

```

b=r1;
do
    compare&swap(r2,b,k); // si r2==k k y b se intercambian
while (b==r3);
r1=1          r2=0          r3=1
    
```