

# WUOLAH



irenychuchu

[www.wuolah.com/student/irenychuchu](http://www.wuolah.com/student/irenychuchu)



11389

## Leccion\_8.pdf

*Apuntes Tema 3*



2º Arquitectura de Computadores



Grado en Ingeniería Informática



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

**CUNEF**

POSTGRADO EN  
**DATA SCIENCE**

Excelencia, futuro, éxito.

 Santander

*Programa Financiación a la  
Excelencia CUNEF-Banco  
Santander e incorporación  
al banco finalizado el máster.*

## TEMA 3:

# ARQUITECTURAS CON PARALELISMO A NIVEL DE THREAD (TLP)

## Lección 2: COHERENCIA DEL SISTEMA DE MEMORIA.

Los sistemas de memoria en multiprocesadores incluyen cachés de todos los nodos, memoria principal, controladores, buffers (uno de escritura/almacenamiento y otro que combinan escrituras/almacenamientos, etc.) y una red de interconexión (medio de comunicación de todos estos componentes).

La comunicación de datos entre procesadores la realiza el sistema de memoria. La lectura de una dirección debe devolver lo último que se ha escrito (desde el punto de vista de todos los componentes del sistema).

### Incoherencia en el sistema de memoria.

- La presencia de cachés en un computador conlleva que pueda haber varias copias de una misma dirección en el sistema de memoria de un computador; en particular, además de estar en memoria principal, puede haber una copia de caché en un procesador, y en el caso de multiprocesadores, en la caché de varios procesadores. Si un procesador escribe en la copia de una dirección de memoria que tiene en su caché, se presenta una situación de incoherencia porque la copia de la dirección en esta caché no coincidiría con el contenido de la dirección en memoria principal, habría entonces dos copias de una dirección en el sistema de memoria con distinto contenido. Habría entonces, una **incoherencia entre caché y memoria principal**. Si, además, hay copia de esa dirección en otras cachés, también se presentaría **incoherencia entre cachés**. No se pueden permitir estas situaciones de incoherencia porque permitirían que dispositivos de E/S o procesadores pudieran acceder a valores no actualizados de una dirección en lugar de acceder a lo último escrito en la misma.
- Se recuerda que las cachés y memoria principal se dividen en bloques de un tamaño igual a la unidad de transferencia entre caché y memoria. Se usa *marco de bloque o línea* de caché para denominar a cada una de estas divisiones en la caché, usando *bloque* para las subdivisiones en memoria principal. Cuando el contenido de una dirección no se encuentra en caché se copia en un marco de bloque de la misma, todo el bloque donde se encuentra esta dirección. Dado que el tamaño de memoria principal es mayor que el tamaño de caché, se debe almacenar información que permita identificar el bloque de memoria que hay en un marco. Esta información se almacena en una tabla o directorio asociado a la caché. En este *directorio de caché* hay una entrada para cada marco de bloque que identifica el bloque de memoria con copia en ese marco e informa de su estado en caché.

Clases de estructuras de datos	Eventos que ponen de manifiesto faltas de coherencia	Tipos de Falta de coherencia
Datos modificables	E/S	Cache-MP
Datos modificables compartidos	Fallo de cache	Cache-MP
Datos modificables privados	Emigra thread/proceso → Fallo cache	Cache-MP
Datos modificables compartidos	Lectura de cache no actualizada	Cache-Cache

**Métodos de actualización de memoria principal implementados en cachés:**

Las situaciones de incoherencia se deben abordar, bien no permitiendo que se produzcan o bien evitando que causen problemas en caso de permitirse. Actualmente las cachés implementan dos mecanismos de actualización de memoria principal:

- Escritura inmediata o directa (write-through): Cada vez que se escribe en la copia de un bloque de memoria en una caché, se modifica también el bloque en memoria principal. En las aplicaciones ejecutadas en un computador, normalmente, si se escribe en una dirección de memoria es muy probable que, tras un tiempo reducido, se escriba en direcciones próximas o en la misma dirección (principios de localidad espacial y temporal, respectivamente). Si es así sería más rentable escribir el bloque completo en memoria una vez realizadas las múltiples escrituras en la copia de caché.
- Posescritura (write-back): Se escribe en memoria principal un bloque modificado en una caché cuando se desaloja éste para hacer sitio a otro bloque de memoria al que se acaba de acceder. Así se pueden escribir a la vez en memoria principal múltiples modificaciones de un bloque, y de esta forma se evitan actualizaciones de memoria principal no necesarias.  
Se añade un bit de estado en la entrada del directorio de caché de todos los marcos de bloque que informa de si la copia del bloque del marco es coherente o no con el bloque de memoria.  
El controlador de memoria activará este bit de un marco cuando se modifique por primera vez la copia del bloque que contiene.

Lo usual es que las cachés usen posescritura. Si un computador usa cachés con posescritura se pueden producir situaciones de incoherencia entre caché y memoria principal. Para evitar problemas, se debe detectar cuando un dispositivo o procesador accede a una posición de memoria modificada en la caché de otro dispositivo o procesador y, en este caso, se le debe suministrar la última actualización de la dirección. Todo esto formaría parte del protocolo de mantenimiento de coherencia del sistema de memoria.

**Alternativas para propagar una escritura en protocolos de coherencia de caché:**

Se utilizan dos alternativas para propagar escrituras a otras cachés:

- Escritura con actualización (write-update): Cada vez que un procesador escribe en una dirección en su caché se escribe también en las copias de esa dirección en otras cachés. Si se cumplen los principios de localidad espacial y/o temporal, se puede generar un tráfico innecesario.
- Escritura con invalidación (write-invalidate): Antes de que un procesador modifique una dirección en su caché se invalidan todas las copias del bloque, que contiene la dirección, en otras cachés. Esta alternativa permite que se pueda seguir escribiendo, después de la primera escritura en el bloque, sin generar tráfico. En este caso, se necesita que el estado de un bloque en el directorio de caché pueda informar de si la copia del bloque en el marco se ha invalidado.

La propagación de las escrituras se puede realizar:

- Difusión: Con una difusión de los paquetes de actualización o invalidación a todas las cachés.
- Envío selectivo: Con el envío de los paquetes de actualización o invalidación solo a aquellas cachés con copia del bloque, que son las únicas que realmente necesitan recibirlos.

### Requisitos del sistema de memoria para evitar problemas por incoherencia.

En una estructura UMA se puede utilizar una red bus para las transferencias entre los procesadores y la memoria que comparten. La red bus implementa la difusión de forma natural. Todos los paquetes enviados a través del bus se ven por todos los componentes conectados al bus. Además, el orden en el que llegan las peticiones al bus establece un orden (serializa) los accesos, en particular, las escrituras. Todo esto hace que, en el caso de un multiprocesador UMA con red bus no sea necesario mantener información de las cachés con copias de los bloques y que incluso se pueda suprimir el directorio de memoria principal.

Conclusión:

- Propagar escrituras en una dirección: La escritura debe hacerse visible en un tiempo finito a otros procesadores.
- Serializar las escrituras en una dirección: Las escrituras en una dirección deben verse en el mismo orden por todos los procesadores (el sistema de memoria debe parecer que realiza en serie las operaciones de escritura en la misma dirección).

La red no es un bus:

- Propagar escrituras en una dirección:
  - Usando difusión: los paquetes de actualización/invalidación se envían a todas las caches.
  - Para conseguir mayor escalabilidad: se debería enviar paquetes de actualización/invalidación solo a cachés (nodos) con copia del bloque, y, además, mantener en un directorio, para cada bloque, los nodos con copia del mismo.
- Serializar escrituras en una dirección: El orden en el que las peticiones de escritura llegan a su home (nodo que tiene en MP la dirección) o al directorio centralizado sirve para serializar en sistemas de comunicación que garantizan el orden en las transferencias entre dos puntos.

Directorio de memoria principal. Alternativas para implementarlo:

- Centralizado: compartido por todos los nodos. Contiene información de los bloques de todos los módulos de memoria.
- Distribuido: Las filas se distribuyen entre los nodos. Típicamente el directorio de un nodo contiene información de los bloques de sus módulos de memoria.

### Protocolos de mantenimiento de coherencia. Clasificación y diseño.

Para diseñar o describir un protocolo de mantenimiento de coherencia se debe especificar:

- Política de actualización de memoria principal (escritura directa, posescritura).
- Política de propagación de escrituras de una caché a otras (escritura inmediata, escritura de invalidación).
- Comportamiento:
  - \* Los posibles estados de un bloque: Las acciones del controlador de caché ante eventos que reciba de un bloque dependerán del estado del bloque en caché.
  - \* Los posibles estados de un bloque en memoria principal (o directorio): Las acciones del controlador de memoria principal ante eventos que reciba de un bloque dependerán de este estado del bloque.
  - \* Los paquetes que genera el controlador de caché ante eventos (peticiones de lectura o escritura del procesador asociado a la caché, reemplazo del bloque o paquetes que llegan a través de la red) que reciba referentes a un bloque.

“ El Máster en Data Science de CUNEF es específico para el sector financiero y tiene como elemento diferenciador la combinación de ciencia (modelos y técnicas) y experiencia (conocimiento del negocio de las entidades financieras).”

JUAN MANUEL ZANÓN  
Director - CRM & Commercial  
Intelligence Expert

YGROUP



Convierte el desafío en  
oportunidad y especialízate  
en Data Science.

Más de 1.600  
acuerdos con  
empresas

Arquitectura de computadores

Irene Muñoz Domingo

- \* Los paquetes que genera el controlador de memoria principal ante eventos (paquetes recibidos a través de la red) que reciba referentes a un bloque.
- \* Relación de acciones con eventos y estados.

Clasificación de protocolos para mantener coherencia en el sistema de memoria:

- Protocolos de espionaje (snoopy): Para buses, y en general sistemas con una difusión eficiente (bien porque el número de nodos es pequeño o porque la red implementa difusión).
- Protocolos basados en directorios: Para redes sin difusión o escalables (multietapas y estáticas).
- Esquemas jerárquicos: Para redes jerárquicas (jerarquía de buses, jerarquía de redes escalables, redes escalables-buses).

Facetas de diseño lógico en protocolos para coherencia:

- Política de actualización de MP: escritura inmediata, posescritura, mixta.
- Política de coherencia entre caches: escritura con invalidación, escritura con actualización, mixta.
- Describir comportamiento:
  - Definir posibles estados de los bloques en caché, y en memoria.
  - Definir transferencias (indicando nodos que intervienen y orden entre ellas) a generar ante eventos:
    - \* Lecturas/escrituras del procesador del nodo.
    - \* Como consecuencia de la llegada de paquetes de otros nodos.
  - Definir transiciones de estados para un bloque en caché, y en memoria.

Se van a describir cuatro protocolos de mantenimiento de coherencia, dos para multiprocesadores UMA con red bus (MSI y MESI) y dos para multiprocesador NUMA en una placa (MSI con y sin difusión). Todos ellos usan posescritura, como política de actualización de memoria e invalidación, como política de propagación de escrituras a otras cachés. Las iniciales del nombre de los posibles estados de un bloque en caché forman parte del nombre del protocolo.

### Protocolo MSI de espionaje.

Los protocolos para buses se llaman de espionaje o sondeo porque todos los controladores, al estar conectados al bus, pueden espiar los accesos del resto sondeando el bus. MSI es el protocolo con menor número de estados que utiliza posescritura e invalidación.

Estados de un bloque en caché:

- Modificado (M): un bloque en este estado en caché es la única copia del bloque válida en todo el sistema de memoria. El controlador de caché debe responder con el bloque si le llega una petición con lectura de bloque, y deberá escribirlo en memoria si se reemplaza por otro bloque.
- Compartido (S): un bloque en este estado en caché es válido, también está válido en memoria y puede que haya copia válida en otras cachés. El controlador de caché invalida esta copia del bloque si le llega una petición de invalidación del bloque.
- Inválido (I): Esta físicamente en la caché, pero en estado inválido o bien no está físicamente. Si el procesador de la caché accede a una dirección de este bloque, se genera un fallo de caché y el controlador enviará un paquete de petición del bloque por la red (bus).

POSTGRADO EN DATA SCIENCE

CUNEF

Excelencia,  
futuro, éxito.

pág. 4

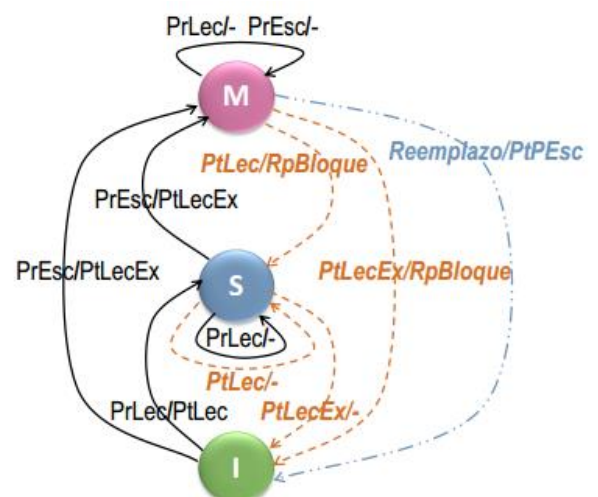
WUOLAH

Estados de un bloque en memoria:

- **Válido:** el bloque está actualizado en la memoria principal y puede haber copia válida en una o varias cachés. El controlador de memoria responde con el bloque si ve en el bus una petición con lectura de un bloque que tiene en este estado.
- **Inválido:** el bloque no está actualizado en memoria principal, hay una copia válida en una caché.

Transferencias generadas por un nodo con caché (tipos de paquetes):

- Son los paquetes que el controlador de caché puede enviar por el bus ante los eventos que se pueden producir en el procesador del nodo (lectura y escritura), en el controlador de caché o que llegan en forma de paquete desde otros nodos. Un controlador de caché genera paquetes de petición (Pt) y de respuesta (Rp).
- **Petición lectura de un bloque (PtLec(B)):** Lectura con fallo de caché del procesador del nodo. El nodo intenta leer un dato en caché, y si no está lo pide a memoria. Como respuesta a esta petición recibirá un paquete de respuesta con el bloque (RpBloque(B)) de memoria o de la caché que lo tiene en estado modificado, si la hubiera.
- **Petición de acceso exclusivo a un bloque con lectura del bloque (PtLecEx(B)):** Consecuencia de una escritura del procesador (evento PrEsc(B)) en un bloque compartido o inválido en la caché. Como respuesta a esta petición se invalidarán las copias del bloque en otras cachés y recibirá un paquete de respuesta con el bloque (RpBloque(B)).
- **Petición de acceso exclusivo a un bloque (PtEx(B)):** Lo genera por escritura del procesador (evento PrEsc). Si el bloque estuviera válido, pero en estado Modificado, no necesita pedir acceso exclusivo porque ya lo tiene. Como respuesta a esta petición se invalidarán las copias del bloque en otras cachés.
- **Petición de posescritura de un bloque (PtPEsc(B)):** Reemplazo de un bloque en estado Modificado.
- **Respuesta con bloque (RpBloque(B)):** Un nodo devuelve un bloque. Se genera cuando tiene un estado Modificado el bloque B solicitado por un paquete con petición de lectura B (PtLec(B) o PtLecEx(B)) recibida.



EST. ACT.	EVENTO	ACCIÓN	SIGUIENTE
<b>Modificado (M)</b>	PrLec/PrEsc		Modificado
	PtLec	Genera paquete respuesta (RpBloque)	Compartido
	PtLecEx	Genera paquete respuesta (RpBloque) Invalida copia local	Inválido
	Reemplazo	Genera paquete posescritura (PtPEsc)	Inválido
<b>Compart. (S)</b>	PrLec		Compartido
	PrEsc <small>posescr</small>	Genera paquete PtLecEx (PtEx)	Modificado
	PtLec		Compartido
	PtLecEx <small>inválido</small>	Invalida copia local	Inválido
<b>Inválido (I)</b>	PrLec	Genera paquete PtLec	Compartido
	PrEsc	Genera paquete PtLecEx	Modificado
	PtLec/PtLecEx		Inválido

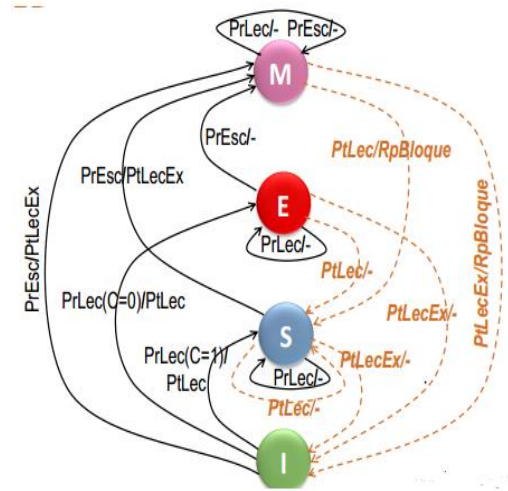
## Protocolo MESI de espionaje.

Con el protocolo MSI siempre que se escribe en la copia de un bloque en una caché se genera un paquete de petición con acceso exclusivo al bloque, aunque no haya copias en otras cachés que se tengan que invalidar. Esto genera un tráfico innecesario, especialmente cuando se ejecutan aplicaciones secuenciales. Para evitar este tráfico, el protocolo MESI divide el estado S de MSI en dos estados:

- Exclusivo (E): no hay copias en otras cachés. Solo está en dicha caché y memoria.
- Compartido (S): lo tienen dos o más cachés y memoria.

Además del cambio mencionado en los estados en caché de un bloque, también hay cambios en las acciones generadas.

Contamos también con un bit que nos indica si es un bit compartido o no. "c" será igual a 1 si el bit es compartido, es decir, si más de un nodo tiene el bloque. Si "c" = 0, nadie tiene el bloque.



### Protocolos MSI basado en directorios con o sin difusión:

Protocolos MSI con posescritura e invalidación para NUMA: uno que difunde las peticiones a todos los nodos y el otro que envía las peticiones solo al nodo que tiene el bloque en el trozo de memoria principal más cercano al nodo. Los protocolos son apropiados para un multiprocesador en una placa.

Se distinguen tres tipos de nodos en la red NUMA:

- Nodo **solicitante** de un bloque (S): Es el nodo que genera una petición del bloque (PtLec, PtEx, PtLecEx o PtEsc).
- Nodo **origen** de un bloque (O): En un multiprocesador NUMA la memoria está repartida físicamente entre los nodos, de forma que cada uno tiene en módulos de memoria más cercanos un trozo del espacio de direcciones total del multiprocesador. En el nodo origen de un bloque es aquél que tiene el bloque en el trozo de memoria que hospeda.
- Nodo **propietario** de un bloque (P): Es un nodo que tiene copia del bloque en su caché.

MSI con directorios sin difusión.

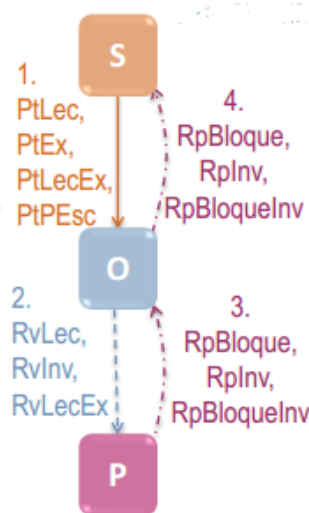
- Como no se usa difusión se necesita almacenar en el directorio de memoria principal, además del estado del bloque en memoria, información sobre las cachés con copia del bloque, para que las invalidaciones se puedan propagar sólo a los nodos con copia del bloque. Se usa un vector de bits de presencia para almacenar esta información. Habría un bit para cada caché conectada con la red NUMA. Un bit activo (a 1), significa que hay copia válida del bloque en esa caché. Todas las peticiones de un bloque se envían al nodo origen del bloque, que es quien tiene el subdirectorío (el directorio se divide en subdirectoríos, uno por cada nodo de procesamiento del computador NUMA) con información sobre el bloque. El nodo origen propagará las invalidaciones a los nodos con copia de un bloque.



- Estados de un bloque de caché:
  - Modificado (M).
  - Compartido (S).
  - Inválido (I).
- Estados de un bloque en memoria:
  - Válido: copias válidas en una o más cachés.
  - Inválido: hay una copia válida en una caché en estado Modificado.
- Paquetes generados por controladores de caché o de memoria y otras acciones:
  - Tipos de nodos:
    - \* Origen (O): Aquel nodo cuyo bloque está físicamente en su memoria.
    - \* Solicitante (S): Cualquier nodo que quiera el bloque esté o no esté en su memoria.
    - \* Propietario (P): Son todos aquellos nodos (procesadores) que tienen una copia válida de ese bloque en caché.

Un nodo puede ser origen, solicitante y propietario a la vez.

- Petición de (Pt):
  - \* **Nodo S a O**: lectura de un bloque (PtLec), lectura con acceso exclusivo (PtLecEx: pides un nodo para modificarlo), petición de acceso exclusivo sin lectura (PtEx: ya tienes el dato), posescritura (PtPEsc).
- Reenvío de petición de (Rv):
  - \* **Nodo O a P**: invalidación (RvInv: reenvío con invalidación), lectura (RvLec y RvLecEx ambos son de reenvío de lectura).
- Respuesta de (Rp):
  - \* **Nodo P a O / O a S**: respuesta con bloque (RpBloque), respuesta con o sin bloque confirmando invalidación (RpInv, RpBloqueInv).



El nodo solicitante "solicita" al origen y no al propietario ya que puede haber muchos propietarios de un dato. ¿Por qué se solicita al origen? Porque cada memoria tiene su directorio en la que puede consultar quién tiene el dato y quién no.

Las entradas del directorio son el número de nodos más la entrada de memoria.



Más de 1.600  
acuerdos con  
empresas

MSI con directorios con difusión:

- Al usar difusión, se envían los paquetes de petición a todas las cachés tengan o no copia del bloque. Por tanto, no se requiere almacenar en el directorio de memoria principal información sobre las cachés con copia del bloque. El protocolo que se va a describir es el más sencillo que se puede implementar con posescritura e invalidación. Los estados de un bloque en caché y en memoria coinciden con los que tiene una implementación sin difusión.

- Estados de un bloque en caché:
  - Modificado (M).
  - Compartido (C).
  - Inválido (I).
- Estados de un bloque en MO:
  - Válido.
  - Inválido.
- Transferencias (tipos de paquetes):
  - Tipos de nodos:
    - \* Origen (O).
    - \* Solicitante (S).
    - \* Propietario (P).

El solicitante (S) cuando pide, habla para todos, es decir, todos se enteran de la petición.

- Difusión de petición del nodo S a:
  - \* **O y P:** lectura de un bloque (PtLec), lectura con acceso exclusivo (PtLecEx), petición de acceso exclusivo sin lectura (PtEx).
  - \* **O:** posescritura (PtPEsc).
- Respuesta de:
  - \* **Nodo P a O:** respuesta con bloque (RpBloque); respuesta con o sin bloque confirmando inválido (RpInv, RpBloqueInv).
  - \* **Nodo O a S:** respuesta con bloque (RpBloque), respuestas con o sin bloque confirmando fin inválido (RpInv, RpBloqueInv).

amazon

McKinsey & Company

KPMG

accenture

pwc

Morgan Stanley

CUNEF

Excelencia,  
futuro, éxito.