

Uso de Redes Heterogéneas para Mejorar la Eficiencia Energética de la Coherencia Directa en Many-Core CMPs

Alberto Ros, Ricardo Fernández-Pascual and Manuel E. Acacio ¹

Resumen— El protocolo de coherencia directa DiCo-CMP ha sido propuesto recientemente como una alternativa a los protocolos basados en directorio para el mantenimiento de la coherencia de las cachés en arquitecturas many-core CMP. A diferencia de estos últimos, en DiCo-CMP es la caché que ha de proporcionar el bloque de datos en un fallo (caché propietaria u *owner*) la encargada de almacenar la información de directorio actualizada para dicho bloque así como de ordenar los diferentes accesos al bloque por parte de distintos núcleos. De esta forma, los fallos de caché en DiCo-CMP son enviados directamente a la caché propietaria, evitando así la indirección que supone el tener que acceder a la información de directorio. A través del uso de un mecanismo de pistas (*hints*) se garantiza una alta tasa de éxito en las predicciones de propietario a costa de incrementar de forma notable el tráfico de red, y como consecuencia, la cantidad de energía consumida por la misma. En este trabajo demostramos cómo, a través del uso de una red heterogénea construida con enlaces de dos tipos, la cantidad de energía asociada a los mensajes de pistas se reduce de forma drástica. Al final, esto se traduce en una mejora importante de la eficiencia energética de DiCo-CMP.

I. INTRODUCCIÓN Y ANTECEDENTES

La mayoría de los procesadores multinúcleo (*chip-multiprocessors* o CMPs) actuales emplean el paradigma de memoria compartida tradicional como abstracción de bajo nivel y pueden ser catalogados, por lo tanto, como multiprocesadores de memoria compartida. En estas arquitecturas los núcleos de procesamiento se comunican mediante operaciones de carga y almacenamiento a un espacio de direcciones compartido. Además, con el fin de reducir la latencia media de cada acceso a memoria así como el número total de accesos a la misma, cada núcleo de procesamiento dispone de uno o varios niveles de cachés privadas. La presencia de cachés privadas, aunque beneficiosa, puede conllevar a la aparición de incoherencias en los valores de los datos que cada núcleo de procesamiento observa y prácticamente todos los procesadores multinúcleo actuales (desde los empleados en servidores hasta los que utilizan algunos dispositivos móviles) resuelven el problema implementando un protocolo de coherencia de cachés a nivel hardware que hace que las cachés privadas sean «transparentes» al software. Esta tendencia se espera que siga dándose en los procesadores many-core futuros con decenas de núcleos de procesamiento [1], [2].

Por otro lado, lo más probable es que estos proce-

sadores many-core se diseñen como arrays de celdas idénticas (o casi idénticas) conectadas por medio de una red directa conmutada. Cada una de estas celdas (*tile*) contiene al menos un núcleo de procesamiento, cachés y una conexión a la red on-chip. Estas arquitecturas proporcionan una solución escalable a la hora de gestionar la complejidad del diseño y permiten utilizar de forma efectiva los recursos disponibles en tecnologías VLSI avanzadas.

Conforme el número de núcleos de procesamiento aumenta, el protocolo de coherencia de caché adquiere un papel protagonista, determinando el rendimiento y el consumo de energía de todo el CMP. Los protocolos basados en directorio fueron concebidos para resolver el problema de la coherencia de las cachés en entornos en los que no se disponía de una red totalmente ordenada (como un bus compartido) [3]. Este es el caso de los many-core CMPs, que emplean una red directa conmutada que no permite una ordenación total de los mensajes que circulan por ella. En un many-core CMP la información de directorio se distribuye normalmente entre los bancos de la caché del último nivel (la L2 en este trabajo). Este nivel de caché suele ser compartido entre todos los núcleos y la información de directorio se suele incluir en la parte de las etiquetas (si se mantiene la inclusividad con respecto a las caches privadas) o en una cache de directorio aparte (en caso contrario) [4]. De esta forma, cada celda contiene la información de compartición correspondiente a los bloques de memoria mapeados al banco de L2 que forma parte de ella, y a este banco de L2 se le denomina origen (*home*) de estos bloques. Aunque los protocolos basados en directorio permiten escalar la coherencia de cachés más allá de los límites impuestos por los protocolos basados en fisgoneo (*snooping*), la indirección que supone el acceso al directorio (que está en el camino crítico de cada fallo de caché) incrementa la latencia de los fallos y consecuentemente merma las prestaciones obtenidas.

Han surgido más recientemente propuestas alternativas a los protocolos basados en directorio para el mantenimiento de la coherencia de las cachés en arquitecturas many-core CMP. Entre ellas se encuentra el protocolo de coherencia directa DiCo-CMP [5]. En DiCo-CMP, la caché que ha de proporcionar el bloque de datos en un fallo (caché propietaria u *owner*) es la encargada de almacenar la información de directorio actualizada para dicho bloque así como de ordenar los diferentes accesos al bloque por parte de distintos núcleos. De esta forma, los fallos de caché

¹Dpto. Ing. y Tecnología de Computadores, Universidad de Murcia, e-mail: {aros,rfernandez,meacacio}@ditec.um.es

en DiCo-CMP son enviados directamente a la caché propietaria, evitando así la indirección que supone el tener que acceder a la información de directorio. Para poder tener información actualizada sobre qué caché es en cada momento la propietaria de un bloque de datos, DiCo-CMP emplea una estructura denominada caché L1 de coherencia (L1C\$) que se mantiene actualizada a través de un mecanismo de pistas (*hints*). Estas pistas son mensajes de control que se envían a todas las L1C\$ en cada cambio de propietario, notificando la nueva identidad. Aunque el uso del mecanismo de pistas garantiza una alta tasa de éxito en las predicciones de propietario, también incrementa de forma notable el tráfico de red, y como consecuencia, la cantidad de energía consumida por la misma.

En este trabajo abordamos este problema que presenta el protocolo DiCo-CMP, y demostramos cómo puede ser resuelto mediante la utilización de una red heterogénea con enlaces de dos tipos. Nuestra propuesta parte de dos observaciones sobre las características de los mensajes de pistas que ocasionan el problema: la primera observación es que estos mensajes no están en el camino crítico del fallo de caché que los genera; la segunda es que estos mensajes son cortos. Basándonos en estas dos observaciones proponemos utilizar una red heterogénea con un conjunto de enlaces de bajo consumo, ancho de banda y coste para transmitir los mensajes de pistas. El resto de mensajes irían por enlaces normales. A través del empleo de esta red la cantidad de energía asociada a los mensajes de pistas se reduce de forma drástica. Al final, esto se traduce en una mejora importante de la eficiencia energética de DiCo-CMP. Finalmente, se comprueba que un protocolo basado en directorio tradicional difícilmente puede verse beneficiado del empleo de esta red de interconexión [6]. Esto es así debido a que la gran mayoría de los mensajes enviados por un protocolo de directorio están en el camino crítico de los fallos de caché. Este resultado puede comprobarse en la Fig. 1, que muestra los mensajes enviados en un protocolo de directorio y en DiCo-CMP clasificados en mensajes críticos y no críticos. Como puede observarse, en media más de la mitad de los mensajes enviados en DiCo-CMP son no críticos (53,5%). Este número se reduce de forma significativa cuando consideramos un protocolo basado en directorio (29,5%).

II. USO DE LA RED HETEROGÉNEA PARA LOS PROTOCOLOS DE COHERENCIA

Los mensajes intercambiados entre los nodos de coherencia se pueden clasificar según su criticalidad. Se considera que un mensaje es crítico si su latencia afecta de forma directa a la latencia de un fallo de caché. Es decir, cuando se encuentra en el camino crítico de una transacción de coherencia que resuelve el fallo de caché.

Una vez identificados qué mensajes del protocolo son críticos y cuales no, se puede aprovechar una red heterogénea para reducir el consumo medio por

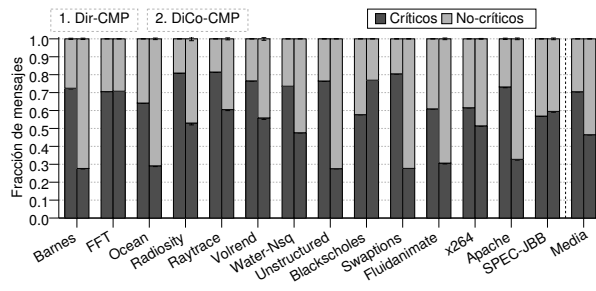


Fig. 1. Comparativa del ratio de mensajes críticos en Dir-CMP y DiCo-CMP.

mensaje sin incrementar la latencia de los fallos ni el tráfico de la red [7]. La clave está en observar que los mensajes fuera del camino crítico se pueden enviar por enlaces de bajo consumo aunque estos enlaces tengan una latencia mayor siempre y cuando el incremento de latencia no sea tan grande como para que el mensaje llegue «demasiado tarde».

Si un mensaje llega demasiado tarde puede afectar al rendimiento aunque no incremente la latencia del fallo de caché que lo originó. Esto puede ocurrir por dos razones:

- El mensaje tiene que llegar antes de que se pueda tratar un fallo de caché distinto (posterior) al que lo originó. Por ejemplo, en la mayoría de protocolos de directorio si la caché de directorio recibe dos peticiones consecutivas a la misma dirección no podrá atender la segunda de ellas hasta que reciba un mensaje de finalización (denominado UNBLOCK) de la primera, a pesar de que este mensaje está fuera del camino crítico del fallo que originó la primera petición. En cierto sentido, el mensaje de UNBLOCK originado por el primer fallo no es crítico para ese fallo pero puede convertirse en crítico para el segundo fallo. A estos mensajes los denominamos *indirectamente críticos*.
- La información del mensaje puede quedar obsoleta o se puede perder una oportunidad de aprovecharla. Por ejemplo, en DiCo-CMP, si la información transportada por los mensajes de pistas llega demasiado tarde normalmente disminuirá la precisión de las predicciones. Sin embargo, dado que el protocolo utiliza esos mensajes sólo para actualizar información de predicción que, por definición, puede ser inexacta, ningún nodo esperará nunca la llegada de uno de estos mensajes antes de atender cualquier petición. A estos mensajes los denominamos *no críticos*.

Hay que observar que incluso aunque un mensaje se considere no crítico no se puede aumentar su latencia indefinidamente sin afectar al tiempo de ejecución. Si se pudiera, significaría que dicho mensaje no tiene utilidad real para el protocolo.

Por tanto, un protocolo podrá reducir el consumo debido al tráfico de red utilizando una red heterogénea con mayor efectividad cuantos más mensajes no críticos envíe y cuanto más se pueda incrementar la

latencia de estos mensajes sin que afecte al tiempo de ejecución.

En el caso de los protocolos considerados en este trabajo, hemos clasificado los mensajes según se describe a continuación:

- Protocolo Dir-CMP:
 - **Mensajes críticos:** Todos los mensajes de peticiones excepto los relacionados con los reemplazos, y todas las respuestas excepto los mensajes de UNBLOCK y el mensaje final de un reemplazo (que incluye datos). Por tanto, este grupo incluye mensajes de peticiones enviadas por los nodos (tanto de lectura como de escritura), peticiones reenviadas, mensajes de respuesta con datos, mensajes de invalidación, y respuestas a las invalidaciones.
 - **Mensajes indirectamente críticos:** Los mensajes de finalización de transacción (UNBLOCK) y los mensajes de finalización de un reemplazo (WRITEBACK_DATA). En ambos casos estos mensajes tienen que llegar al directorio (situado en el último nivel de caché) antes de que éste pueda atender a la siguiente petición para la dirección implicada.
 - **Mensajes no críticos:** Mensajes de iniciación de postescritura. El protocolo Dir-CMP realiza postescritura de L1 a L2 en tres saltos: primero se envía un mensaje de iniciación (PUT) al cual la caché L2 contesta con un mensaje de reconocimiento (ACK) cuando está lista para aceptar los datos (normalmente después de realizar otro reemplazo a memoria). El mensaje de iniciación no es crítico ni indirectamente crítico porque en caso de retrasarse el directorio y la L2 podrían seguir atendiendo peticiones para la dirección implicada. Las nuevas peticiones serían reenviadas al nodo que necesita hacer el reemplazo y cuando el mensaje PUT llegara a la L2 sería contestado con un reconocimiento negativo (NACK).
- Protocolo DiCo-CMP:
 - **Mensajes críticos:** Al igual que en el caso del protocolo de directorio, todos los mensajes de petición, reenvío y respuesta excepto los relacionados con los reemplazos y el mensaje de UNBLOCK son críticos.
 - **Mensajes indirectamente críticos:** En DiCo-CMP los reemplazos se realizan con un solo mensaje que lleva los datos. Este mensaje es indirectamente crítico, de igual forma que el mensaje de finalización de reemplazo de Dir-CMP. Además, DiCo-CMP utiliza un par de mensajes (CHANGE_OWNER y un ACK) para informar al nodo *home* de qué nodo es el propietario actual del bloque y, por tanto, posee la información de compartición sobre el mismo. Estos mensajes también son indirectamente críticos, puesto que algunos fallos no pueden ser contestados por el *owner* hasta que reciba el ACK. En DiCo-CMP no se envía mensaje de finalización de transacción de

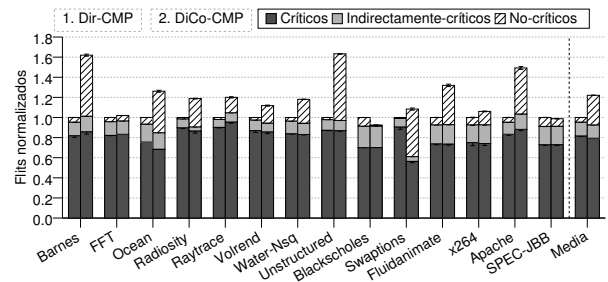


Fig. 2. Comparativa del tráfico de red de Dir-CMP y DiCo-CMP según su criticidad.

coherencia (UNBLOCK).

- **Mensajes no críticos:** DiCo-CMP utiliza mensajes de pistas (*hints*) que no son críticos. Estos mensajes sirven para actualizar la información de predicción almacenada en la LIC\$, que como es información de predicción no es necesario que sea correcta para que el protocolo funcione correctamente. Un mensaje de *hint* que se retrase no provoca que ninguna petición tarde más en enviarse o responderse, aunque puede provocar que el mecanismo de predicción falle y la petición sea enviada a un nodo que no es el propietario actual del bloque, lo cual implicaría que la petición sería reenviada al nodo *home* y el fallo podría tardar más en resolverse finalmente debido a la indirección, además de incrementar el tráfico.

La Fig. 2 muestra el tráfico de red de ambos protocolos clasificado por la criticidad de los mensajes. Se puede observar que aunque el tráfico total de DiCo-CMP es superior al del directorio, todo el tráfico extra se debe a mensajes no críticos mientras que la cantidad de mensajes críticos de DiCo-CMP es la misma o incluso ligeramente menor que en el caso de Dir-CMP.

Esta distribución del tráfico sugiere que DiCo-CMP obtendrá un mayor beneficio que DiCo-CMP de la utilización de una red de interconexión heterogénea. En efecto, como se muestra a continuación una red heterogénea va a permitir aproximar el consumo debido a la red de interconexión de DiCo-CMP al de Dir-CMP (que requiere menos tráfico) sin reducir la ventaja en el tiempo de ejecución de DiCo-CMP sobre Dir-CMP.

III. RED HETEROGÉNEA CON ENLACES DE BAJO CONSUMO

Trabajos previos han mostrado que una parte importante del consumo global de energía en un CMP se debe a los enlaces punto a punto de la red de interconexión [8], [7]. Por otro lado, es posible diseñar enlaces con diferentes características de latencia, ancho de banda o consumo variando el número de hilos, la anchura de los hilos, el espaciado entre hilos, la separación entre repetidores y el tamaño de estos; y es posible combinar varios tipos de enlaces en la misma red utilizando cada tipo de enlace para transportar

Tipo de red	Enlaces	Enlaces de bajo consumo (por mensaje)		Área extra
		Consumo dinámico relativo	Latencia relativa	
Base (Homogénea)	128 B-Wires	—	—	0 %
Heterogénea ×4	128 B-Wires y 32 PW-Wires	0.3×	4×	25 %
Heterogénea ×8	128 B-Wires y 16 PW-Wires	0.3×	8×	12,5 %
Heterogénea ×16	128 B-Wires y 8 PW-Wires	0.3×	16×	6,25 %

TABLA I
CONFIGURACIONES DE RED

diferentes tipos de mensajes según las necesidades de latencia y ancho de banda de cada tipo, obteniendo reducciones en el tiempo de ejecución y consumo para algunas topologías de redes indirectas [6], [9]. Posteriormente, se propuso en [7] una red heterogénea simplificada con dos tipos de enlaces: enlaces de baja latencia para mensajes críticos y enlaces de bajo consumo para mensajes no críticos. Esta red permite obtener reducciones en el consumo y el tiempo de ejecución si se combina con una técnica de particionado de mensajes de respuesta con datos que permite separar todos los mensajes en dos tipos: mensajes críticos y pequeños y mensajes no críticos y grandes.

En este trabajo también proponemos una red heterogénea con dos tipos de enlaces: enlaces normales y enlaces de bajo consumo. Los enlaces de bajo consumo que proponemos incrementan poco el área necesaria para implementar la red de interconexión porque no sólo utilizan hilos de menor anchura y menor número de repetidores a cambio de mayor latencia, sino que se utiliza un número menor de hilos por enlace a cambio de menor ancho de banda. A diferencia de trabajos anteriores, nuestra red heterogénea está optimizada para transmitir otros dos tipos de mensajes: mensajes críticos (incluyendo mensajes grandes) y mensajes no críticos y pequeños.

Los enlaces de nuestra red de interconexión están compuestos de 2 tipos de hilos de los 4 propuestos en [6]: B-Wires que utilizamos para la red base y PW-Wires que utilizamos en la red de bajo consumo y que tienen el doble de latencia, consumen un 70 % menos por cada bit transmitido y ocupan el mismo área. Utilizando estos dos tipos de hilos, proponemos tres configuraciones para la red heterogénea que se muestran en la TABLA I. En estas configuraciones se varía el número de hilos añadidos para la red de bajo consumo. En el primer caso (Heterogénea ×52) se utilizan la mitad de PW-Wires que de B-Wires dado que la red de bajo consumo está optimizada para el envío de mensajes pequeños (pistas y mensajes de control de 8 bytes). El resto de configuraciones reducen aún más el número de PW-Wires, reduciendo el ancho de banda a cambio de requerir menor área.

IV. ENTORNO DE EVALUACIÓN

La evaluación de Dir-CMP y DiCo-CMP sobre la red heterogénea propuesta en la sección anterior se ha realizado mediante el simulador funcional Virtutech Simics [10] extendido con Multifacet GEMS 2.1 [11]. GEMS modela la jerarquía de memoria y ofrece a Simics la latencia de acceso a memoria de

TABLA II
Parámetros del sistema

Parámetros de la memoria	
Jerarquía de caché	No inclusiva
Tamaño de bloque	64 bytes
Caché L1 de datos e instr.	64KB, 4 vías
Latencia de acceso a L1	2 ciclos
Caché L2 compartida	512KB/celda, 8 vías
Latencia de acceso a L2	2 (etiq.) y 6 (datos) ciclos
Caché de directorio	Ilimitada
Caché de coherencia L1	1KB, 4 vías, 2 ciclos
Caché de coherencia L2	1KB, 4 vías, 2 ciclos
Tiempo de acceso a memoria	160 ciclos
Parámetros de la red	
Topología	Malla 2-D (4×4)
Técnica de enrutamiento	Determinista X-Y
Tamaño de flit	16 bytes
Tamaño de mensajes	5 flits (datos), 1 flit (control)
Tiempo de routing y switch	2 y 2 ciclos
Enlaces rápidos	2 ciclos
Enlaces de bajo consumo	4, 8 y 16 ciclos

cada petición del procesador. La implementación de la red heterogénea se ha realizado sobre el simulador GARNET [12], incluido en GEMS. La arquitectura simulada corresponde a un tiled CMP con 16 núcleos. Los principales parámetros de evaluación se muestran en la TABLA II. Nótese que el hecho de tener una cache de directorio ilimitada beneficia únicamente al protocolo basado en directorio, ya que DiCo-CMP no posee cachés de directorio, sino las llamadas cachés de coherencia las cuales consideramos limitadas.

Para la evaluación hemos usado aplicaciones de diversas suites de programas de prueba: *Barnes* (16K partículas), *FFT* (64K complejos), *Ocean* (océano 514×514), *Radiosity* (room, -ae 5000.0 -en 0.050 -bf 0.10), *Raytrace* (teapot -optimizada tras eliminar cerrojos innecesarios-), *Volrend* (head) y *WaterNsq* (512 moléculas) pertenecen a la suite SPLASH-2 [13]. *Unstructured* (Mesh.2K) es una aplicación científica irregular. *Blackscholes* (simmedium), *Fluidanimate* (simmedium), *Swaptions* (simmedium) y *x264* (simmedium) pertenecen a la suite PARSEC [14]. Finalmente, *Apache* (1000 transacciones HTTP) y *SPEC-JBB* (1600 transacciones) son dos aplicaciones comerciales [15]. Todos los resultados mostrados en este trabajo son de la parte paralela de las aplicaciones evaluadas.

V. RESULTADOS

En esta sección estudiamos el rendimiento y consumo de energía de los protocolos basados en directorio (Dir-CMP) y los basados en coherencia directa (DiCo-CMP) sobre una red heterogénea que usa

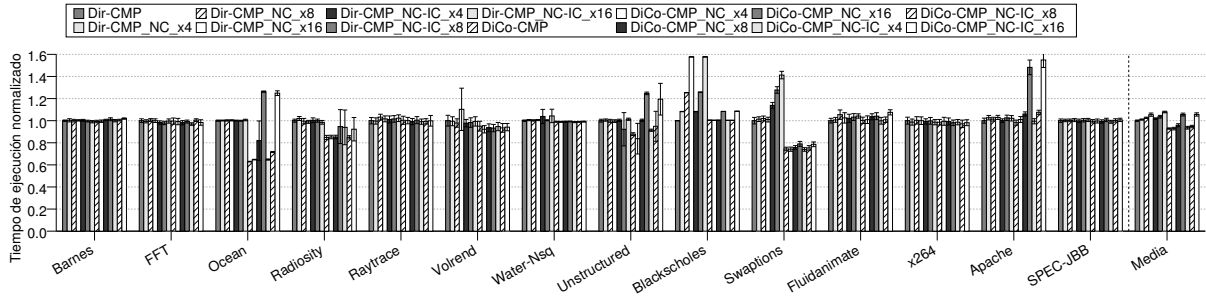


Fig. 3. Tiempo de ejecución para Dir-CMP y DiCo-CMP con varias latencias para enlaces de bajo consumo.

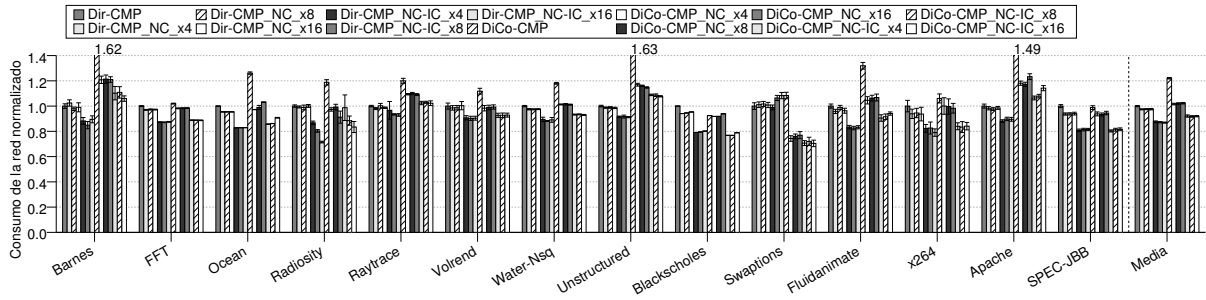


Fig. 4. Consumo de la red para Dir-CMP y DiCo-CMP con varias latencias para enlaces de bajo consumo.

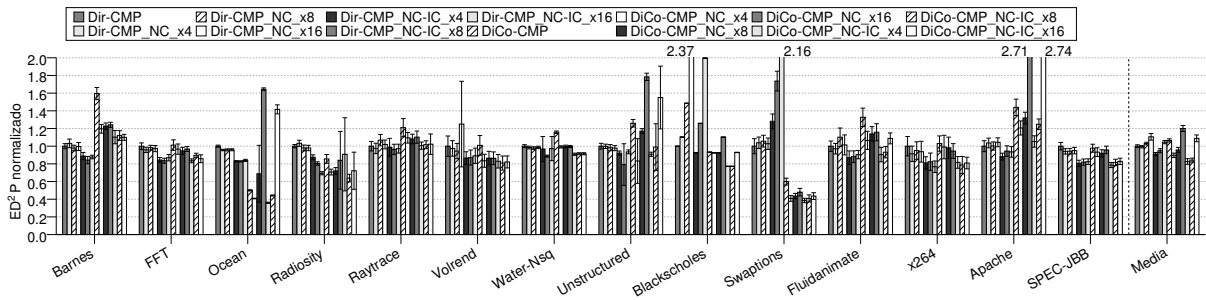


Fig. 5. ED^2P para Dir-CMP y DiCo-CMP con varias latencias para enlaces de bajo consumo.

dos tipos de enlaces: *B-Wire* y *PW-Wire*. En cuanto al consumo de energía nos centramos en la energía dinámica. Más concretamente, estudiamos dos situaciones. En la primera se analiza el impacto que tiene enviar todos los mensajes por los enlaces *B-Wire* (configuración base) con respecto a enviar solamente los mensajes no críticos por los enlaces *PW-Wire* (configuración *NC*) o enviar tanto mensajes no críticos como indirectamente críticos por los *PW-Wires* (configuración *NC-IC*). En la segunda se analiza diversas latencias para los enlaces *PW-Wire* respecto a los *B-Wire*. Por ejemplo, $\times 4$ indica que la latencia de los enlaces *PW-Wire* es cuatro veces mayor que la de los *B-Wire*. En nuestro estudio, el aumento de la latencia implica una reducción del tamaño de *flit* y por tanto del número de hilos de los enlaces *PW-Wire*, como se mostró en la TABLA I.

La Fig. 3 muestra el tiempo de ejecución para las distintas configuraciones evaluadas normalizado con respecto a Dir-CMP. En primer lugar podemos observar que DiCo-CMP es capaz de reducir el tiempo de ejecución respecto a Dir-CMP en un 7,1 % de

media. Al enviar mensajes no críticos (o incluso los indirectamente críticos) por los *PW-Wires* el tiempo de ejecución aumenta ligeramente para los dos protocolos. Este incremento se hace más notable al aumentar la latencia de dichos enlaces (reduciendo el número de hilos). De hecho, es en DiCo-CMP donde este aumento se puede apreciar más, en especial para una latencia $\times 16$, donde ya iguala las prestaciones de Dir-CMP. Esto se debe a que la llegada demasiado tardía a su destino de los mensajes de pistas se traduce en un aumento de los fallos de la predicción del propietario del bloque de memoria solicitado. En general, DiCo-CMP tolera bastante bien una latencia de hasta $\times 8$ para mensajes no críticos, lo cual permite una reducción del consumo manteniendo una sobrecarga de área de red pequeña (12,5 %).

Gracias al envío de mensajes no críticos por enlaces de bajo consumo, el consumo de la red se puede ver reducido en gran medida. La Fig. 4 muestra la energía dinámica consumida por la red para las diferentes configuraciones, normalizada respecto a Dir-CMP. Podemos ver que, de media, DiCo-CMP con-

sume un 22 % más que Dir-CMP en la configuración base debido al aumento de tráfico por los mensajes de pistas. Al enviar estos mensajes por los *PW-Wires* su consumo se reduce en un 70 %. De este modo el consumo de DiCo-CMP se reduce hasta un 17 % de media. Si además consideramos los mensajes indirectamente críticos, el consumo se reduce hasta un 25 %, llegando a consumir solo un 5 % más que un protocolo de directorio que también envíe mensajes no críticos e indirectamente críticos por enlaces de bajo consumo. Nótese que en este estudio sólo consideramos el consumo dinámico y no el estático o de fuga (*leakage*). El consumo estático se reduciría al reducir el número de hilos usados por los enlaces de bajo consumo.

Finalmente, la Fig. 5 muestra el valor obtenido por cada una de las configuraciones para la métrica ED^2P (energía por retardo al cuadrado), de nuevo normalizado respecto a Dir-CMP. Para esta métrica podemos observar la importancia de enviar mensajes indirectamente críticos por los enlaces de bajo consumo. De hecho, Dir-CMP casi no se ve afectado por el envío de los mensajes no críticos por los *PW-Wires*, ya que éstos son bastante escasos. Para la configuración en la que se envían también los mensajes indirectamente críticos, se obtiene una mejora de un 8,9 % respecto a Dir-CMP para los enlaces $\times 4$. Sin embargo, DiCo-CMP supera esta mejora en ED^2P con sólo enviar mensajes no críticos por los *PW-Wires* (10,3 %). Si además enviamos los mensajes indirectamente críticos por este tipo de enlaces obtenemos una mejora en ED^2P de 17,3 % para enlaces $\times 4$ con respecto a la configuración base y de un 9,7 % respecto a la mejor configuración de Dir-CMP (*Dir-CMP_NC-IC_ $\times 4$*). Estos resultados demuestran que los protocolos basados en coherencia directa pueden aprovechar las redes heterogéneas mejor que los protocolos basados en directorio.

VI. CONCLUSIONES

En este artículo hemos mostrado como los protocolos de coherencia directa se pueden beneficiar mucho más del uso de redes heterogéneas que los protocolos de directorio tradicionales. En particular, la red estudiada tiene únicamente dos tipos de enlaces: normales y de bajo consumo. Gracias al envío de mensajes que no son críticos por enlaces de bajo consumo, se logra una importante reducción en el consumo dinámico de la red, que constituye una parte importante del consumo total del chip.

Como el número de mensajes que no son críticos en los protocolos de coherencia directa (53,5 %) es muy superior al de los basados en directorio (29,5 %), los primeros pueden hacer un mayor uso de las redes de bajo consumo. Ya que el aumento en la latencia de los mensajes no críticos no repercute de forma significativa en el tiempo de ejecución, esta importante reducción en energía se puede obtener con un coste en rendimiento casi despreciable.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación (MICINN) y la Comisión Europea FEDER mediante los proyectos “Consolider Ingenio-2010 CSD2006-00046” y “TIN2009-14475-C04-02”. Agradecemos a Pablo David Muñoz Sánchez el trabajo realizado en el simulador usado en este trabajo durante su proyecto fin de carrera.

REFERENCIAS

- [1] Jacob Leverich, Hideho Arakida, and Alex Solomatnikov, et al, “Comparing memory systems for chip multiprocessors,” in *34th Int'l Symp. on Computer Architecture (ISCA)*, June 2007, pp. 358–368.
- [2] Milo M.K. Martin, Mark D. Hill, and Daniel J. Sorin, “Why on-chip cache coherence is here to stay,” *To appear in Communications of the ACM*, 2012.
- [3] David E. Culler, Jaswinder P. Singh, and Anoop Gupta, *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann Publishers, Inc., 1999.
- [4] Michael Zhang and Krste Asanović, “Victim replication: Maximizing capacity while hiding wire delay in tiled chip multiprocessors,” in *32nd Int'l Symp. on Computer Architecture (ISCA)*, June 2005, pp. 336–345.
- [5] Alberto Ros, Manuel E. Acacio, and José M. García, “DiCo-CMP: Efficient cache coherency in tiled CMP architectures,” in *22nd Int'l Parallel and Distributed Processing Symp. (IPDPS)*, Apr. 2008, pp. 1–11.
- [6] Liquan Cheng, Naveen Muralimanohar, Karthik Raman, Rajeev Balasubramanian, and John B. Carter, “Interconnect-aware coherence protocols for chip multiprocessors,” in *33rd Int'l Symp. on Computer Architecture (ISCA)*, June 2006, pp. 339–351.
- [7] Antonio Flores, Juan L. Aragón, and Manuel E. Acacio, “Heterogeneous interconnects for energy-efficient message management in cmps,” *IEEE Transactions on Computers (TC)*, vol. 59, no. 1, pp. 16–28, Jan. 2010.
- [8] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shamir, “Interconnect-power dissipation in a microprocessor,” in *Int'l workshop on System Level Interconnect Prediction (SLIP)*, Feb. 2004, pp. 7–13.
- [9] Kaustav Banerjee and Amit Mehrotra, “A power-optimal repeater insertion methodology for global interconnects in nanometer designs,” *IEEE Transactions on Electron Devices*, vol. 49, no. 11, pp. 2001–2007, Nov. 2002.
- [10] Peter S. Magnusson, Magnus Christensson, and Jesper Eskilson, et al, “Simics: A full system simulation platform,” *IEEE Computer*, vol. 35, no. 2, pp. 50–58, Feb. 2002.
- [11] Milo M.K. Martin, Daniel J. Sorin, and Bradford M. Beckmann, et al, “Multifacet’s general execution-driven multiprocessor simulator (GEMS) toolset,” *Computer Architecture News*, vol. 33, no. 4, pp. 92–99, Sept. 2005.
- [12] Niket Agarwal, Tushar Krishna, Li-Shiuan Peh, and Niranjan K. Jha, “GARNET: A detailed on-chip network model inside a full-system simulator,” in *IEEE Int'l Symp. on Performance Analysis of Systems and Software (ISPASS)*, Apr. 2009, pp. 33–42.
- [13] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta, “The SPLASH-2 programs: Characterization and methodological considerations,” in *22nd Int'l Symp. on Computer Architecture (ISCA)*, June 1995, pp. 24–36.
- [14] Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li, “The PARSEC benchmark suite: Characterization and architectural implications,” in *17th Int'l Conference on Parallel Architectures and Compilation Techniques (PACT)*, Oct. 2008, pp. 72–81.
- [15] Alaa R. Alameldeen, Carl J. Mauer, Min Xu, Pacia J. Harper, Milo M.K. Martin, Daniel J. Sorin, Mark D. Hill, and David A. Wood, “Evaluating non-deterministic multi-threaded commercial workloads,” in *5th Workshop On Computer Architecture Evaluation using Commercial Workloads (CAECW)*, Feb. 2002, pp. 30–38.