Análisis de Esquemas de Memoria Caché para Ambientes con Múltiples Hilos de Ejecución

José Luis Hamkalo[1], Augusto J. Vega[1] y Bruno Cernuschi-Frías[2]

[1] Facultad de Ingeniería, Universidad de Buenos Aires, Argentina
jhamkal@fi.uba.ar
[2] Facultad de Ingeniería, Universidad de Buenos Aires y CONICET, Argentina

Resumen Se analizan las organizaciones de memoria caché SWSA-MT y asociativas por conjuntos de 2 y 4 vías para procesadores multihilo. Se utiliza un nuevo modelo para la clasificación de desaciertos en memoria cache para ambientes multihilo, denominado "modelo de las 4C'. El modelo discrimina los desaciertos de conflicto como "cerrados" (un hilo consigo mismo) y "cruzados" (entre hilos). Los resultados obtenidos, a partir de la ejecución de los *benchmarks* SPLASH-2, muestran un desempeño comparable entre las organizaciones SWSA-MT y 4WSA y superiores a la 2WSA. Se observa que la caché SWSA-MT presenta una menor tasa de desaciertos por conflicto cruzado, en relación a las otros esquemas estudiados. Esta es la virtud más sobresaliente de la organización SWSA-MT ya que, al disponer de memorias privadas para los hilos, se minimiza la interferencia "destructiva" entre los mismos.

1. Introducción

Las memorias cache han jugado un rol central en el aumento sostenido del desempeño de las computadoras de altas prestaciones. Las mejoras en la arquitectura del procesador y la tecnología han contribuido en forma pareja durante las dos décadas pasadas como las fuerzas principales para conseguir niveles sin precedentes en el funcionamiento de procesadores de propósito general. Los procesadores de hoy en día utilizan "pipelines" muy profundos y tecnologías de integración nanométricas decrecientes. Esto resulta en tasas de reloj muy altas, ejerciéndose así una alta presión sobre el sistema de memoria. Esta presión demanda un muy alto grado de eficiencia en el subsistema de memoria cache, siendo necesario continuas mejoras, las cuales se basan en nuevos esquemas, estrategias y tecnologías.

Los procesadores de propósito general de alto desempeño se basan principalmente en una micro arquitectura superescalar. Para maximizar el aprovechamiento de los recursos de dichos procesadores se requiere un alto grado de paralelismo a nivel de instrucciones (ILP - Instruction Level Parallelism). En la práctica estos procesadores se caracterizan por desperdiciar capacidad de cómputo, permaneciendo durante algunos ciclos de reloj en estado ocioso. En ocasiones, la ejecución del flujo de instrucciones suele bloquearse debido a, entre otras cosas,

malas predicciones de los saltos, desaciertos en la memoria cache de instrucciones, operaciones de E/S, etc., lo que se traduce indefectiblemente en un procesador ocioso, a la espera de una instrucción a ejecutar.

Existen dos formas de generación tiempo ocioso, la primera para algunas de las unidades funcionales y la segunda para todo el procesador en si mismo y han sido llamadas:

- **Desperdicio Horizontal**: debido al bajo nivel de ILP que impide, para un ciclo de reloj, usar eficientemente los recursos (unidades funcionales) disponibles.
- **Desperdicio Vertical**: debido al bloqueo o latencias en las instrucciones ejecutadas, con lo cual, el procesador debe permanecer ocioso.

Una solución al uso ineficiente de los recursos del procesador consiste en la utilización de múltiples hilos (o flujos) de instrucciones, técnica conocida como *ejecución multihilo*.

En los procesadores con multihilo explícito existen las siguientes alternativas para el manejo concurrente de la ejecución:

- **FMT** (*Fine-grain Multithreading*): los hilos se alternan en cada instrucción ejecutada.
- **CMT** (*Coarse-grain Multithreading*): los hilos se alternan cuando el que estaba en ejecución incurre en alguna penalidad, tal como un desacierto en la memoria caché L2 [9].
- **SMT** (*Simultaneous Multithreading*): consiste en ejecutar dos o mas hilos en forma simultánea.

La técnica SMT, propuesta por Eggers y otros en [5], busca minimizar tanto el desperdicio horizontal como también el desperdicio vertical. Para evitar que el procesador permanezca ocioso, las instrucciones a ejecutar se toman de varios flujos simultáneamente, de manera tal que, ante el bloqueo de uno de ellos, se pueda continuar con la ejecución de alguno de los otros. Un esquema SMT busca maximizar, para cada ciclo de reloj, la utilización de los recursos del procesador. Si el flujo en ejecución presenta un alto nivel de ILP, entonces ese paralelismo permite explotar al máximo la utilización de las unidades funcionales para un ciclo de reloj. Por otra parte, si varios flujos presentan cada uno un bajo nivel de ILP, entonces pueden ser ejecutados simultáneamente, para maximizar el aprovechamiento de los recursos. El uso de la técnica SMT genera nuevos desafíos, particularmente en el diseño de las unidades funcionales que se ven involucradas y afectadas por la explotación del paralelismo a nivel de hilo (TLP − *Thread Level Parallelism*). Es decir, debe replantearse la concepción de aquellos recursos que se comparten entre los hilos. La memoria caché de nivel 1 es una de las componentes más afectadas, debido a que la existencia de múltiples instrucciones pertenecientes a múltiples hilos de ejecución dentro del procesador genera una situación de *competencia* que se traduce en una alta interferencia de los hilos en las líneas de la memoria caché. Por ejemplo, para un esquema de

Correspondencia Directa (DM - *Direct Mapped*) y una carga dada, la tasa de desaciertos (*miss rate*), desde uno a ocho hilos, va desde 2.5 % a 14.1 % (para la caché de instrucciones), y desde 3.1 % a 11.3 % (para la caché de datos) [16].

Este trabajo tiene como principal objetivo realizar una contribución a través de la propuesta de una herramienta que permita comprender cuales son las posibles causas de los desaciertos en memoria cache. En este sentido es que se propone un nuevo modelo para clasificar los desaciertos en memoria cache en ambientes con múltiples hilos de ejecución. Se llama al modelo propuesto modelo de las cuatro C o 4C. Se dan las definiciones operativas correspondientes, las cuales se basan en la utilización de las distancias de pila LRU. El modelo 4C clasifica a las referencias en memoria de manera individual, formando una taxonomía, siendo posible el análisis de la pertenencia de una referencia a memoria entre las diferentes categorías. El modelo de las 4C propuesto se aplica en el presente trabajo a organizaciones de memoria cache estandar y la organización SWSA-MT recientemente propuesta [8].

2. Clasificación de Desaciertos

En el estudio de organizaciones de memoria caché, una de las figuras de mérito es la tasa de desaciertos (*miss rate*). Es de interés conocer el motivo por el cual se producen los desaciertos en la caché, de manera tal de poder tomar decisiones en el diseño de la organización de memoria. En este sentido, se han propuesto varias clasificaciones, siendo la más popular la perteneciente a M. Hill et al. [11], conocida como "modelo de las 3C". Este modelo clasifica los desaciertos en tres tipos: obligatorios (*compulsory*), de capacidad (*capacity*) y de conflicto (*conflict*). Cada una de estas clases es cuantificada de la siguiente forma:

- Desaciertos obligatorios (*compulsory*): son los producidos en una organización de memoria caché de tamaño infinito. Dicho de otra manera, son los desaciertos debidos al acceso por vez primera a bloques de memoria.
- Desaciertos de capacidad (*capacity*): son los desaciertos no obligatorios, producidos en una organización LRU completamente asociativa (*fully-associative*) de igual tamaño a la caché siendo estudiada. Básicamente, son causados por referenciar mayor cantidad de bloques que los que caben en la caché.
- Desaciertos de conflicto (*conflict*): son los producidos en la caché siendo estudiada, menos los desaciertos de capacidad y obligatorios.

El "modelo de las 3C" es ampliamente usado en el estudio de memorias caché. Sin embargo, es sabido que en algunos casos puede generar una cantidad negativa de desaciertos de conflicto. Teniendo en cuenta la definición brindada en el párrafo anterior, esto significa que existen referencias que son aciertos en la caché siendo estudiada pero son desaciertos en la organización completamente asociativa de igual tamaño. Estos desaciertos son denominados "de anticonflicto" [13] y, si bien se consideran poco frecuentes en la práctica [10] [13], Hamkalo et. al [6] demuestran que su presencia puede ser significativa en programas de

evaluación típicos, como los *benchmarks* SPEC95. Por ejemplo, el promedio de los SPECfp95 para una caché de correspondencia directa de 8-KB, produce una tasa de desaciertos de anticonflicto del 21,1 % (para el caso de instrucciones), y una tasa del 8,3 % (para el caso de datos). Para el promedio de los SPECint95, una caché de 16-KB de instrucciones presenta una tasa de desaciertos de anticonflicto del 19,7 %, mientras que la de datos presenta una tasa del 9,9 %.

Hamkalo et. al [6] proponen una extensión al "modelo de las 3C", denominado "modelo de las 3C determinístico" (D3C), y que evita la existencia de cantidades negativas de desaciertos a partir de la generación de una taxonomía. En el modelo D3C, la clasificación se redefine de la siguiente manera:

- Desaciertos obligatorios (*compulsory*): son los producidos en una organización de memoria caché de tamaño infinito, y también en la caché siendo estudiada[3].
- Desaciertos de capacidad (*capacity*): son los desaciertos no obligatorios, producidos en la caché siendo estudiada, y también en una organización LRU completamente asociativa (*fully-associative*) de igual tamaño.
- Desaciertos de conflicto (*conflict*): son los producidos en la caché siendo estudiada, menos los desaciertos de capacidad y obligatorios.

Para la clasificación anterior puede darse una definición operativa a partir del concepto de "distancia LRU" para un bloque B. Dicha distancia es definida como la cantidad de bloques diferentes que fueron referenciados desde el último acceso a B. De esta forma, el desacierto producido por referenciar a un bloque B con distancia LRU D en una caché de T bloques de tamaño, puede clasificarse de la siguiente manera:

- Obligatorio $\rightarrow$ distancia D no computable
- Capacidad $\rightarrow D > T$
- Conflicto $\rightarrow D \leq T$

La redefinición en los conceptos de desacierto obligatorio y de capacidad, evita computar como desaciertos a aquellas referencias que, en la práctica, son aciertos en la caché estudiada. Como consecuencia, el modelo D3C nunca genera componentes negativas de desaciertos y, además, permite construir una taxonomía al clasificar individualmente cada referencia.

Como se analiza en la sección 4, los modelos 3C y D3C presentan dificultades para clasificar patrones de accesos en ambientes multihilo y, por esta razón, allí se replantea la clasificación.

3. Adaptación del esquema SWSA a un ambiente multihilo: SWSA-MT

La organización SWSA como fue concebida originalmente [7] no evita el problema de la interferencia entre los hilos que se ejecutan, dado que los ban-

[3] Este requisito adicional contempla los casos en donde un bloque es referenciado por primera vez, pero no es desacierto por haber sido leído, a la caché estudiada, en forma adelantada mediante algún mecanismo de *prefetching*.

cos no son privados, sino que se comparten entre todos los hilos de ejecución. En realidad, tampoco lo evitan otras organizaciones típicas, a menos que cada hilo disponga de una memoria para uso privado, lo cual no es admisible (por el desperdicio que se originaría en escenarios con menor cantidad de hilos disponibles). En [15] se efectúa un análisis respecto del impacto en el desempeño de una memoria caché DM de nivel 1, privada y compartida. Se deduce que un esquema privado por hilo de ejecución tiene un mal desempeño para pocos o un único hilo, ya que en ese caso se desperdician porciones de memoria que no pueden ser accedidas. Como contrapartida, un esquema compartido se desempeña muy bien con pocos hilos en ejecución, pero se hace muy evidente la interferencia cuando estos aumentan en número.

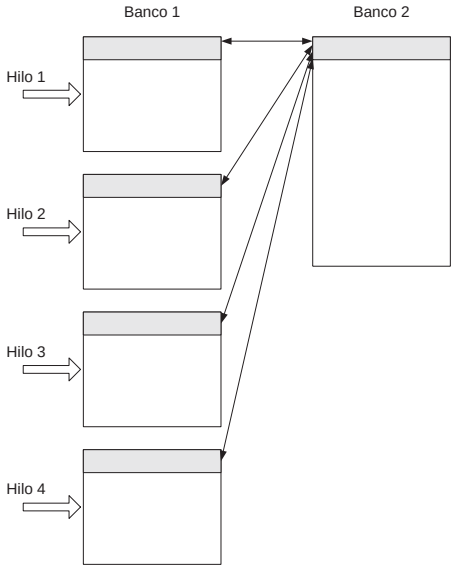

Figura 1: Esquema de asociatividad en la organización SWSA-MT.

La organización que se presenta a continuación es una adaptación del esquema SWSA), donde el primer banco es privado (por hilo de ejecución), mientras que el segundo banco se comparte entre todos los hilos (ver figura 1). El acceso a esta memoria se lleva a cabo mediante indexación por bits, como en las organizaciones caché tradicionales. Además, SWSA-MT permite tamaños totales que no sean potencia de 2, como sucede en el diseño SWSA.

Si ante un desacierto en su memoria privada y en la memoria compartida, un hilo x encuentra el dato en la memoria privada de otro hilo y (siendo $x \neq y$), entonces estamos en presencia de un acierto "largo". Ya que el dato en cuestión fue accedido tanto por x como por y, se lo considera "compartido", y se lo

reubica en el banco común[4]. A esta estrategia se la denominó "reubicación" (*reallocation*) y, en adelante, se utilizarán como sinónimos los términos SWSA-MT, SWSA-MT con reubicación, SWSA-MT *realloc*, para hacer referencia a la organización estudiada.

4. El Modelo de las 4C

Los modelos de clasificación de desaciertos, 3C y D3C, presentados en la sección 2 fueron concebidos para ambientes de un único flujo de ejecución. La existencia de múltiples hilos genera interrogantes sobre la aplicabilidad de dichos modelos, y presenta desafíos adicionales que deben ser analizados. Además, como se mostró en la sección 3, cada hilo tiene acceso a una parte acotada de la caché, conformada por su porción privada más la vía compartida. Entonces, no sería correcto clasificar un desacierto como de capacidad utilizando como parámetro el tamaño total de la memoria, siendo que cada hilo solo puede hacer un uso parcial de la caché. Por estas razones, en este trabajo se replanteó la clasificación convencional, y se generó un nuevo modelo, llamado "de las 4C", y que es aplicable en ambientes multihilo.

En primer lugar, es de interés discriminar los desaciertos de conflicto en un ambiente multihilo, según:

- Conflictos propios (un hilo consigo mismo).
- Conflictos entre hilos (interferencia).

Esta subclasificación no está contemplada en los modelos 3C y D3C, y es de particular importancia para entender en qué grado interfieren los hilos en la organización caché estudiada.

Considerando un ambiente como el mostrado en la figura 1, con porciones privadas P y compartida C, la nueva clasificación se define de la siguiente manera:

- Desaciertos obligatorios (*compulsory*): son los producidos en una organización de memoria de tamaño infinito, y también en la caché siendo estudiada, considerando todos los hilos [5].
- Desaciertos de capacidad (*capacity*): son los desaciertos no obligatorios, producidos en la caché siendo estudiada, y también en una organización LRU completamente asociativa (*fully-associative*) de tamaño $P + C$; es decir, la distancia LRU del bloque referenciado es $D > P + C$.
- Desaciertos de conflicto cerrado (*closed-conflict*): son los producidos en la caché siendo estudiada, por referencias a bloques con distancia LRU $D \leq P + C$, y que fueron desalojados por el propio hilo.

[4] El dato que se hallaba en el banco compartido es desalojado sin considerar su antigüedad.

[5] Este requisito es importante, ya que un hilo podría traer a caché un bloque requerido por otro hilo, en cuyo caso no sería un desacierto obligatorio.

- Desaciertos de conflicto cruzado (*crossed-conflict*): son los producidos en la caché siendo estudiada, por referencias a bloques con distancia LRU $D \leq P + C$, y que fueron desalojados por otros hilos.

El "modelo de las 4C" es una extensión del modelo D3C y, por lo tanto, contempla los escenarios convencionales (un hilo, caché tradicional).

5. Ambiente de Simulación

5.1. Herramientas

Para la simulación de las organizaciones caché bajo estudio, se utilizó la herramienta ad hoc SimiOO, mediante la técnica de *simulación manejada por trazas*. El simulador fue "alimentado" con las trazas recolectadas usando la herramienta Valgrind [17].

5.2. Arquitecturas Simuladas

Se simularon tres esquemas de memoria caché de datos de nivel 1: 2WSA, 4WSA y SWSA-MT, de tamaños 8, 16, 32 y 64-KB, línea de 32 bytes, y se consideró un ambiente con 4 hilos de ejecución. La política de reemplazo adoptada, en todos los casos, fue LRU (*Least Recently Used*).

5.3. Benchmarks

Para la simulación de las organizaciones citadas se utilizó un subconjunto de los *benchmarks* SPLASH-2 [19]: CHOLESKY, FFT, LU, OCEAN, RADIX y WATER. SPLASH-2 está construido en base a un conjunto de macros, conocidas como PARMACS, y desarrolladas por ANL (*Argonne National Laboratory*) [4] [14]. Originalmente, SPLASH-2 fue concebido para procesamiento paralelo y, siendo que en este trabajo se requerían *benchmarks* multihilo, se utilizó una reimplementación de las macros PARMACS con soporte para hilos POSIX (*pthreads*), provistas por Ernest Artiaga y otros [1], de la Universidad Politécnica de Cataluña. Todos los *benchmarks* fueron ejecutados sobre Valgrind [17], de donde se extrajeron las trazas de referencias a memoria mediante el módulo Tracegrind.

6. Resultados de las Simulaciones

6.1. Tasa de Desaciertos

Los primeros resultados a analizar corresponden a la tasa de desaciertos para las organizaciones 2WSA, 4WSA y SWSA-MT (con reubicación). Se busca minimizar la cantidad de desaciertos, ya que estos conllevan penalidades muy elevadas, que producen un aumento en la relación ciclos por instrucción (CPI), apartándolo del ideal de 1 ciclo/instrucción.

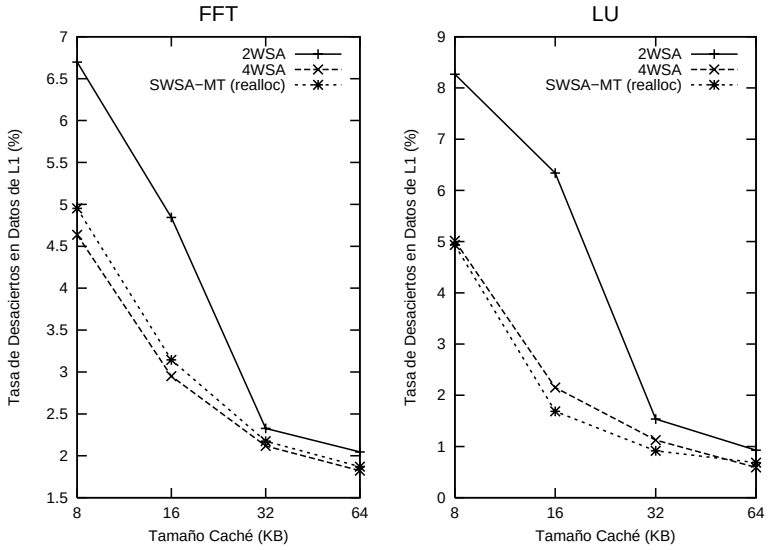

Figura 2: Tasa de desaciertos para los *benchmarks* FFT y LU.

En los resultados que se muestran en las figuras 2 y 3 puede observarse que la organización propuesta SWSA-MT presenta una tasa de desaciertos significativamente inferior a la mostrada por el esquema 2WSA, y muy similar a la del esquema 4WSA.

Las caídas abruptas en la tasa de desaciertos para la organización 2WSA se deben a efectos de hiperpaginación (*thrashing*), por el cual, bloques de memoria caché son continuamente desalojados, independientemente de cuán recientes sean. El fenómeno de *thrashing* puede darse, por ejemplo, en cachés cuyo tamaño no es suficiente para mantener temporalmente el conjunto de datos con los que se está trabajando (*working set*), o también, por escenarios de interferencia "destructiva" entre varios hilos, cuyos *working sets* corresponden a la misma región de la memoria caché. El último caso es el que aplica a los resultados mostrados; es decir, hiperpaginación por conflictos entre hilos. Si se tratase de *thrashing* por capacidad, entonces la caída abrupta en las curvas también se obervaría en las otras organizaciones estudiadas. Este análisis prematuro quedará expuesto a continuación, utilizando el "modelo de las 4C" para clasificación de desaciertos.

6.2. Clasificación de Desaciertos

Para clasificar los desaciertos se utilizó el "modelo de las 4C" elaborado en la sección 4. Los resultados se presentan en las figuras 4, 5 y 6.

En primer lugar, queda en evidencia lo que se comentó en la sección anterior: el punto de inflexión en la tasa de desaciertos para el esquema 2WSA está dado

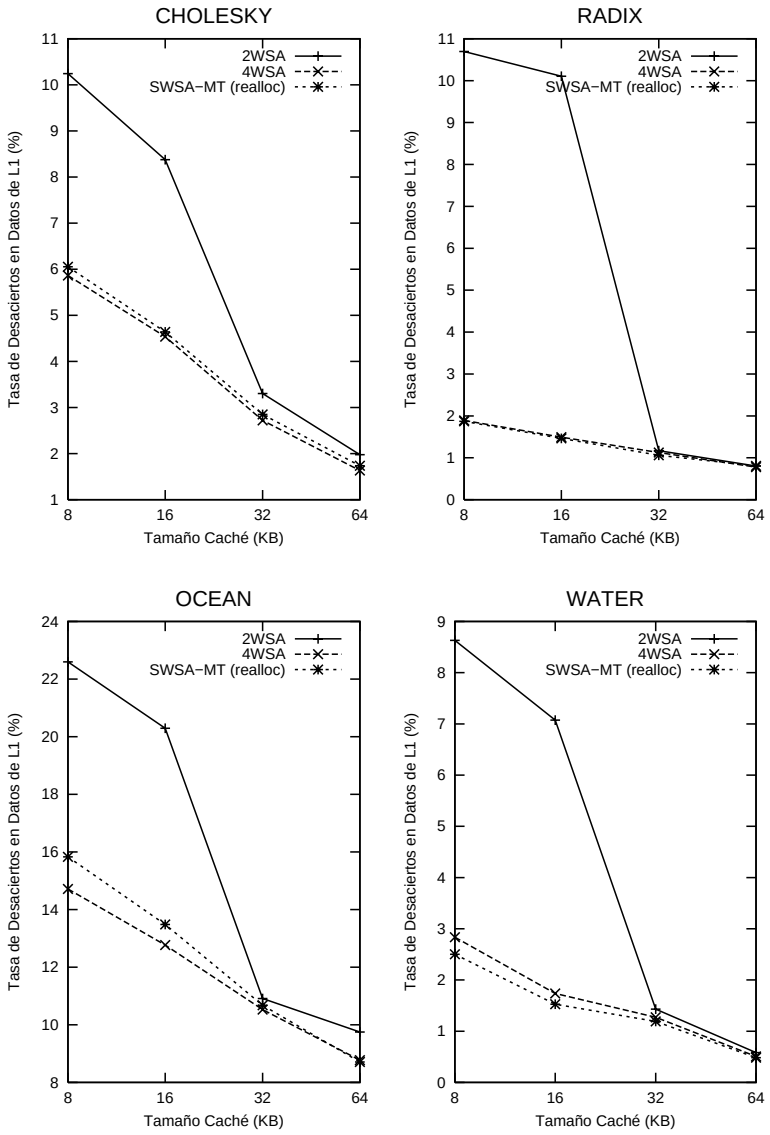

Figura 3: Tasa de desaciertos para los *benchmarks* CHOLESKY, RADIX, OCEAN y WATER.

por una elevada interferencia entre hilos ("conflicto cruzado") para tamaños 8 y 16-KB. Por otra parte, se observa una de las características más importantes de la organización SWSA-MT; en todos los casos, la caché propuesta presenta una menor tasa de desaciertos por conflicto cruzado, en relación a las otros esquemas

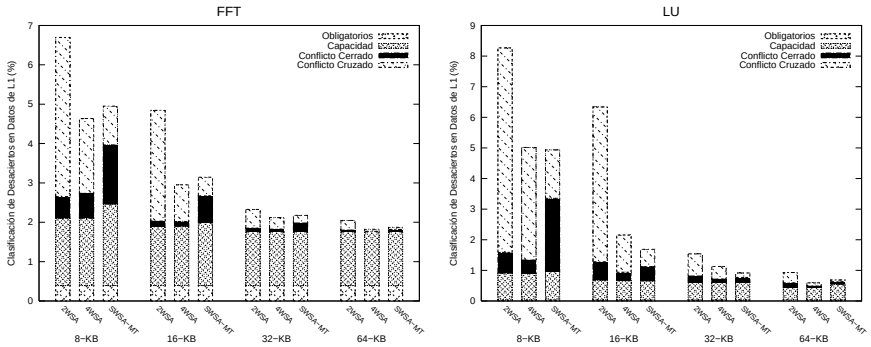

Figura 4: Clasificación de desaciertos para los *benchmarks* FFT y LU.

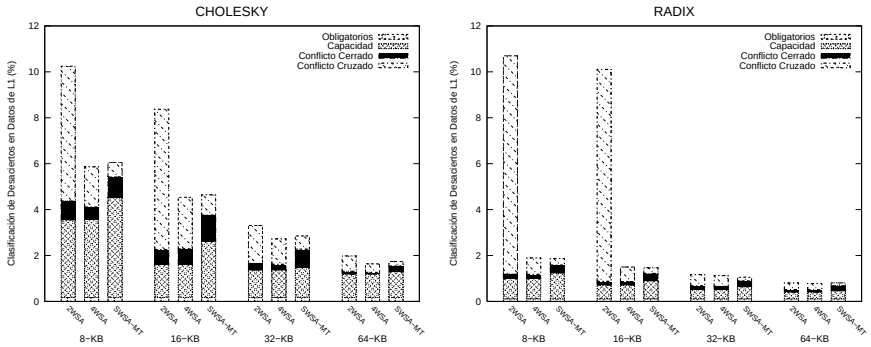

Figura 5: Clasificación de desaciertos para los *benchmarks* CHOLESKY y RADIX.

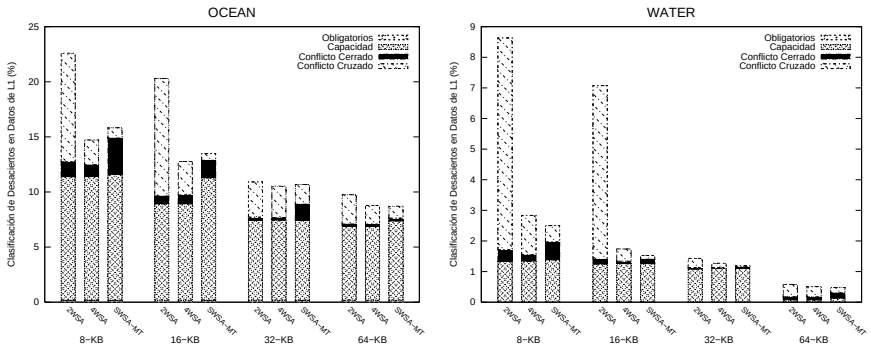

Figura 6: Clasificación de desaciertos para los *benchmarks* OCEAN y WATER.

estudiados. Esta es la virtud más sobresaliente de la organización SWSA-MT ya que, al disponer de memorias privadas para los hilos, se minimiza la interferencia "destructiva" entre los mismos. También es cierto que, casi siempre, la organización propuesta muestra una mayor proporción de conflictos cerrados (un hilo consigo mismo) y de conflictos por capacidad. Esto sucede, por ejemplo, para los tamaños 8 y 16-KB para todos los *benchmarks*, y se debe a que cada hilo puede hacer un uso parcial de la caché ($P + C$ bloques) respecto a los esquemas 2WSA y 4WSA en donde, virtualmente, cada hilo tiene disponible el tamaño total de la memoria. A pesar de estos desaciertos adicionales en el esquema SWSA-MT, éste presenta una tasa de desaciertos comparable a la organización 4WSA y menor a la 2WSA.

7. Conclusiones

Se analizaron las organizaciones de memoria caché SWSA-MT y asociativas por conjuntos de 2 y 4 vías para procesadores multihilo. Los resultados obtenidos, a partir de la ejecución de los *benchmarks* SPLASH-2, muestran un desempeño comparable entre las organizaciones SWSA-MT y 4WSA y superiores a la 2WSA.

Se observa que la caché SWSA-MT presenta una menor tasa de desaciertos por conflicto cruzado, en relación a las otros esquemas estudiados. Esta es la virtud más sobresaliente de la organización SWSA-MT ya que, al disponer de memorias privadas para los hilos, se minimiza la interferencia "destructiva" entre los mismos. Esto hace de SWSA-MT una excelente opción para ser implementada en procesadores multihilo, como los Intel Pentium 4 Hyper-Threading, o la nueva arquitectura SUN UltraSPARC T1 con tecnología CoolThreads.

Además, se propuso un nuevo modelo para la clasificación de desaciertos para ambientes multihilo, denominado "modelo de las 4C" (y que es una extensión del popular "modelo de las 3C"). Este modelo, que se utilizó para la clasificación de desaciertos en el estudio de las organizaciones caché presentadas, es capaz de discriminar los desaciertos de conflicto como "cerrados" (un hilo consigo mismo) y "cruzados" (entre hilos), y contempla el hecho de que los hilos podrían acceder parcialmente al total de la caché, factor que puede resultar se sumo interés en el estudio de nuevos esquemas de memoria cache con porciones de memoria asignadas a los hilos en forma dinámica y reconfigurable en tiempo de ejecución.

8. Agradecimientos

El presente trabajo cuenta con subsidios de la Universidad de Buenos Aires y el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

Referencias

1. E. Artiaga, N. Navarro, X. Martorell, Y. Becerra, "Implementing PARMACS Macros for Shared Memory Multiprocessor Environments," Technical Report, Polytechnic University of Catalunya, Department of Computer Architecture, 1997.

2. D. Bailey, "FFT's in External or Hierarchical Memory," Journal of Supercomputing, pp. 23-35, 1990.
3. G. Blelloch, C. Leiserson, B. Maggs, C. Plaxton, S. Smith, M. Zagha, "A Comparison of Sorting Algorithms for the Connection Machine CM-2," Proc. of the Symposium on Parallel Algorithms and Architectures, pp. 3-16, 1991.
4. J. Boyle, R. Butler, T. Disz, B. Glickfeld, E. Lusk, R. Overbeek, J. Patterson, R. Stevens, "Portable Programs for Parallel Processors," Holt, Rinehart, and Winston Incorporation, New York, NY, 1987.
5. S. Eggers, J. Emer, H. Levy, J. Lo, R. Stamm, D. Tullsen, "Simultaneous Multithreading: A Platform for Next-Generation Processors," IEEE Micro, pp. 12-19, 1997.
6. J. Hamkalo, B. Cernuschi Frías, "A Taxonomy for Cache Memory Misses," Proc. of the 11th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'99), 1999.
7. J. Hamkalo, B. Cernuschi Frías, "Non Symmetric Two Way Set Associative Caches," Proc. of the AST2000, pp. 1-8, 2000.
8. J. Hamkalo, A. Vega, B. Cernuschi Frías, "SWSA-MT: Un Esquema de Memoria Caché para Ambientes Multihilo," 5th Argentine Symposium on Computing Technology (AST 04), 2004.
9. J. Hennessy, D. Patterson, "Computer Architecture. A Quantitative Approach," 3^{ra} edición, Morgan Kaufmann Publishers, 2003.
10. M. Hill, "Aspects of Cache Memory and Instruction Buffer Performance," PhD thesis, Computer Science Division (EECS), University of California, Berkeley, California, USA, November 1987.
11. M. Hill, A. Smith, "Evaluating Associativity in CPU Caches," IEEE Transactions on Computers, pp. 1612-1630, 1989.
12. H. Kwak, B. Lee, A. Hurson, S. Yoon, W. Hahn, "Effects of Multithreading on Cache Performance," IEEE Transactions on Computers, pp. 176-184, 1999.
13. A. Lebeck, D. Wood, "Cache Profiling and the SPEC Benchmarks: A Case Study," IEEE Computer, vol. 27, pp. 15-26, 1994.
14. E. Lusk, R. Overbeek, "Use of Monitors in FORTRAN: A Tutorial on the Barrier, Self-scheduling DO-Loop, and Askfor Monitors," Technical Report No. ANL-84-51, Rev. 1, Argonne National Laboratory, June 1987.
15. D. Tullsen, S. Eggers, H. Levy, "Simultaneous Multithreading: Maximizing On-Chip Parallelism," Proc. of the 22nd Annual International Symposium on Computer Architecture (ISCA-1995), 1995.
16. D. Tullsen, S. Eggers, J. Emer, H. Levy, J. Lo, R. Stamm, "Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor," Proc. of the 23rd Annual International Symposium on Computer Architecture (ISCA-1996), 1996.
17. Valgrind, `http://www.valgrind.org/`
18. S. Woo, J. Singh, J. Hennessy, "The Performance Advantages of Integrating Block Data Transfer in Cache-Coherent Multiprocessors," Proc. of the 6th Symposium on Architectural Support for Programming Languages and Operating Systems, pp. 219-231, 1994.
19. S. Woo, M. Ohara, E. Torrie, J. Singh, A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," Proc. of the 22nd Annual International Symposium on Computer Architecture (ISCA-1995), pp. 24-36, 1995.