

Enseñanza de Arquitecturas de Memorias Caché mediante Simuladores

Miguel A. Vega Rodríguez, Juan A. Gómez Pulido, Juan M. Sánchez Pérez

Departamento de Informática. Universidad de Extremadura
Escuela Politécnica. Campus Universitario, s/n. 10071 Cáceres. Spain
E-mail: mavega@unex.es, jangomez@unex.es, sanperez@unex.es. Fax: +34-927-257202

Resumen

En este trabajo se presentan dos experiencias docentes en la enseñanza de arquitecturas de memoria caché para la titulación de Ingeniería Informática de la Universidad de Extremadura. Estas experiencias se desarrollan desde dos puntos de vista: Sistemas caché multinivel para una asignatura de carácter fundamental ("*Estructura de Computadores*"), y sistemas de memoria caché en multiprocesadores, para una asignatura de carácter avanzado ("*Arquitectura de Computadores*").

1. Introducción

En el plan de estudios de la titulación de Ingeniería Informática de la Universidad de Extremadura (UEX), destacan dos asignaturas troncales, anuales y de carácter fundamental dentro del área de *Arquitectura y Tecnología de Computadores*:

- EC: Estructura de Computadores (2º curso).
- AC: Arquitectura de Computadores (4º curso).

En EC se estudia la memoria caché por primera vez, por lo que es fundamental aquí contar con un conjunto de herramientas docentes que reflejen los conceptos básicos de estas memorias. Con este fin se ha desarrollado el simulador *SISMEC*. En la asignatura AC, partiendo de la base de los conocimientos adquiridos en EC, se amplía el estudio de la memoria caché al caso de los sistemas multiprocesador, para lo cual se desarrolló la herramienta *SMPCache*.

2. SISMEC

SISMEC ha sido desarrollado con el propósito de servir de ayuda en la explicación de los sistemas jerárquicos de memorias caché [3] [4] que se desarrolla en la parte teórica de la asignatura EC durante la primera mitad del segundo cuatrimestre. Para ello, el material para realizar las prácticas es el siguiente:

- *Guión de prácticas*. Es una documentación que explica el manejo del simulador y en la que se exponen las prácticas que hay que desarrollar.
- *Software*. Consiste en el programa simulador *sismec12.exe* (para MS-DOS), y los ficheros de trazas (.prg) suministrados.

Las prácticas han de ser realizadas por el alumno rellenando las indicaciones del guión, el cual se entrega al profesor al finalizar las mismas, con el fin de poder ser evaluado.

2.1. Características

SISMEC es un sencillo simulador, compuesto de una serie de ficheros. En la tabla 1 resumimos las principales características del programa.

Programa	SISMEC12.EXE
Tamaño ejecutable	353 KBytes
Sistema operativo	MS-DOS
Requisitos gráficos	VGA-Color
Entorno	Gráfico

Tabla 1. Principales características de SISMEC

El fichero ejecutable es el simulador, y necesita de los ficheros .prg, que son las trazas de memoria generadas en la ejecución de benchmarks.

Organización

Partimos de la base de una organización de bloques para toda la jerarquía de memoria, de forma que es el bloque la unidad de información referenciable en los niveles. Consideramos que el tamaño de bloque en cualquier nivel es el mismo, así como que el ancho de palabra de memoria (mínima unidad referenciable por los programas) es el mismo en toda la jerarquía. Así, el simulador ofrece las posibilidades resumidas en la tabla 2.

Niveles de caché	Hasta dos niveles
Tipos de caché	Unificada (una caché por nivel), o separada (instrucciones y datos: dos cachés por nivel)
Correspondencia	Directa, asociativa por conjuntos o totalmente asociativa
Reemplazamiento	Aleatorio, LFU, LRU o FIFO
Políticas de escritura	Directa o retardada
Tamaño de palabra	8, 16, 32 ó 64 bits
Palabras por bloque	1, 2, 4, 8, 16, 32, 64, 128 ó 256
Bloques en memoria principal	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 ó 8192
Máx. n° particiones, conjuntos o vías	512

Tabla 2. Configuraciones soportadas por SISMEC

Podemos llegar a configurar bloques de hasta 2 KB, cachés de hasta 1 MB (excluyendo bits de validez, etiquetas, etc.), y memoria principal de hasta 16 MB.

Las distintas configuraciones pueden guardarse en ficheros ASCII (.mem) de datos para una futura carga, evitando así tener que volver a hacer numerosas selecciones para configurar el mismo modelo de memoria.

Uso del simulador

SISMEC 1.2 es la primera versión del simulador. Éste opera sobre una plataforma PC-DOS, y ha sido escrito en C++. Dispone de una interfaz gráfica con la que se pueden modelar sistemas de memoria, seleccionar trazas de prueba, ver resultados finales y en tiempo de ejecución (de forma gráfica y/o numérica), etc. También hay posibilidad de abortar la simulación en cualquier instante, con el objeto de corregir algún detalle de la organización.

Al entrar en el simulador, y tras una pantalla de bienvenida, se ofrecen unos menús con las siguientes opciones:

- **Ficheros.** Se gestionan los ficheros de datos (salvar y cargar), además de permitir la salida e información del programa. **Cargar:** Permite la carga de ficheros de datos: modelos de memoria y trazas de memoria. Éstos se seleccionan de una lista de ficheros que residen en el mismo directorio en el que está ubicado el simulador. **Salvar:** Graba en un fichero .mem el modelo que está configurado en ese momento en el simulador, y cuyo descriptor dará nombre al fichero.

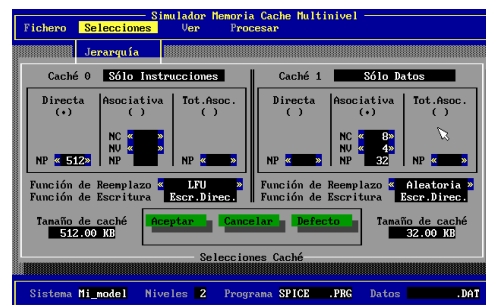


Figura 1. Entorno de selecciones de cachés

- **Seleccionar.** En este menú será configurado, en su totalidad, el modelo de memoria. Para ello, se dispone de las siguientes opciones: **Jerarquía:** Se establecen los principales parámetros de la jerarquía de memoria: ancho de palabra, palabras por bloque y número de bloques de la memoria principal. Con estas tres variables, se establece el tamaño de memoria principal y el de bloque. También, se selecciona el número de niveles de caché y la estrategia de escritura, además del nombre que identificará el modelo. **Niveles:** Selecciona cuántas cachés tiene cada nivel. Obviamente, sólo están permitidas las cachés unificadas y las separadas. **Cachés:** Se establecen los valores de los parámetros que modelan cada caché (ver figura 1). Dependiendo del tipo de correspondencia elegida, se definen unas variables u otras (número de bloques, de conjuntos, de vías). Estas selecciones dan el tamaño de la caché. También, se establece el tipo de estrategia de reemplazo.

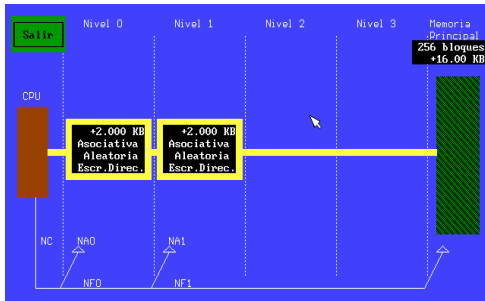


Figura 2. Esquema de un ejemplo de modelo de caché

- *Ver.* Aquí se despliega una ventana en la que aparece el esquema del modelo, junto con sus principales valores (figura 2).

Nivel	Caché	Aciertos	Fallos	Accesos	FA(%)	FF(%)
0	Inst.	0.0	87.0	87.0	0.0	100.0
	Datos	0.0	36.0	36.0	0.0	100.0
1	Inst.	6.0	81.0	87.0	6.9	93.1
	Datos	5.0	31.0	36.0	13.9	86.1
2	Nixta	7.0	107.0	114.0	6.1	93.9
3	Nixta	44.0	65.0	109.0	40.4	59.6
Total:		58.0	65.0	123.0	47.2	52.8

Figura 3. Datos estadísticos en formato texto

- *Procesar.* Se efectúa el análisis y simulación. Los resultados generados consisten en las tasas o frecuencias de aciertos y fallos, tanto globales como individuales de cada caché. Si seleccionamos la opción *campos* (ver figura 3), estos valores serán tabulados numéricamente conforme se va procesando la traza de memoria en el simulador, de forma que se reproduce exactamente la ejecución del programa, analizando su comportamiento en memoria. Se observa dónde se produce cada acierto y cada fallo, se indica qué tipo de operación se está llevando a cabo (captura de instrucción, lectura de dato, escritura de dato) y sobre qué dirección efectiva de memoria. Si seleccionamos la opción *gráficos* (figura 4), se genera la misma simulación, pero omitiendo los valores numéricos, que serán mostrados en forma de gráficos. Tan sólo la frecuencia de fallos

global se muestra de forma numérica.

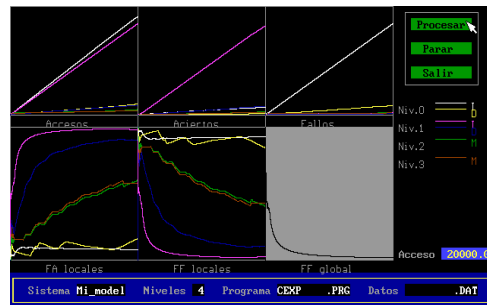


Figura 4. Entorno de gráficos de rendimientos

2.2. Experiencias docentes

Con este simulador se realizan 7 prácticas a lo largo de 7 semanas. Cada práctica tiene una finalidad concreta, con conexión directa con los aspectos abordados en la teoría:

- Poner de manifiesto que los programas tienen distinta localización, y que hay programas con “buena” o “mala” localización.
- Ver la influencia del tamaño de la caché en la tasa de fallos.
- Cachés separadas frente a caché unificada.
- Efecto del aumento del número de niveles de caché.
- Influencia del tamaño de caché como parámetro de diseño en una caché de 2º nivel.
- Influencia del tamaño de bloque.
- Influencia de la función de correspondencia.

Las prácticas son entregadas a los alumnos en unos formularios. Estos contienen una introducción, una descripción de la arquitectura de memoria considerada, las trazas que han de utilizarse, unas tablas cuyos contenidos han de ser obtenidos con los cálculos que efectúa el simulador, una gráfica de resultados a completar y, finalmente, una serie de cuestiones sobre la discusión de los resultados. La experiencia docente indica que se afianzan mejor los conceptos teóricos cuando se realizan estas prácticas, entre otras cosas porque el alumno construye unas gráficas que coinciden con las que aparecen en los libros de texto, al mismo tiempo que observa cómo evolucionan los distintos parámetros de los sistemas de memoria caché a medida que se modifican sus valores.

3. SMPCache

SMPCache es un simulador de sistemas de memorias caché en multiprocesadores simétricos (SMPs) con memoria compartida por bus, que se utiliza para la realización de prácticas en la asignatura AC durante el segundo cuatrimestre. Con él se busca afianzar los conceptos teóricos sobre memorias cachés en multiprocesadores [1] [5], que se imparten en AC. El material para realizar las prácticas es el siguiente:

- *Guiones de prácticas.* Se entregan dos grupos de guiones de prácticas. El primero incluye la documentación que explica el manejo de SMPCache y expone los experimentos que el alumno debe desarrollar utilizando el simulador. Por otro lado, los estudiantes deben construir una versión reducida del simulador. Para ello, se les da un segundo grupo de guiones. Estos explican, paso a paso, todas las operaciones y algoritmos que los alumnos necesitan conocer para construir un “mini-simulador” similar a SMPCache.
- *Software.* Consiste en el programa simulador smpcache.exe (para Windows), y los ficheros de ayuda asociados (.hlp). También se incluye un conjunto inicial de ficheros de trazas (.prg) y de configuración (.cfg).

3.1. Características

Resumimos en la tabla 3 las principales características del programa.

Programa	SMPCACHE.EXE
Tamaño ejecutable	650 KBytes
Sistema operativo	WINDOWS
Requisitos gráficos	SVGA-Color (800x600)
Entorno	Gráfico (tipo Windows)
Otras facilidades	Incluye sistema de ayuda contextual e instalación-desinstalación automática

Tabla 3. Principales características de SMPCache

Organización

El simulador ofrece las posibilidades presentadas en la tabla 4. Podemos llegar a configurar bloques de hasta 8 KB, cachés de hasta 16 MB (excluyendo bits de estado, etiquetas, campos de cómputo, etc.), y memoria principal de 32 GB.

Niveles de caché	1
Tipos de caché	Unificada
Correspondencia	Directa, asociativa por conjuntos o totalmente asociativa
Reemplazamiento	Aleatorio, LFU, LRU o FIFO
Políticas de escritura	Retardada (post-escritura), por protocolos de coherencia
Tamaño de palabra	8, 16, 32 ó 64 bits
Palabras por bloque	1, 2, 4, 8, 16, 32, 64, 128, 256, 512 ó 1024
Bloques en memoria principal	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152 ó 4194304
Particiones, conjuntos o vías	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 ó 2048
Procesadores	1, 2, 3, 4, 5, 6, 7 u 8
Protocolos de coherencia caché	MSI, MESI o DRAGON
Arbitración del bus	Aleatoria, LRU o LFU

Tabla 4. Configuraciones soportadas por SMPCache

Al igual que ocurría en SISMEC, las distintas selecciones que se hacen para configurar el simulador pueden guardarse en ficheros ASCII (.cfg) para una futura carga. Más aún, es posible establecer una configuración inicial por defecto. Estas características permiten construir una base de datos con las diferentes arquitecturas de memoria, pudiendo emular arquitecturas como Silicon Graphics, Sequent Symmetry,...

Uso del simulador

SMPCache 1.0 opera sobre una plataforma PC-WINDOWS, y ha sido escrito mediante un lenguaje visual. El simulador nos ofrece la respuesta del sistema al someterlo a los accesos de memoria generados por los programas (trazas de memoria que se hayan cargado en los distintos procesadores). Permitiendo una visión completa de la evolución del multiprocesador (ver figura 5), de una caché en particular o incluso de un determinado bloque de memoria (figura 6). Mostrando, en cada instante, los accesos a memoria realizados por cada procesador, el estado del bus, de cada caché, de cada bloque,...

Algunos de los factores que se pueden estudiar con el simulador son: Localidad de los distintos programas; influencia del número de procesadores, del protocolo de coherencia caché, del algoritmo de arbitración del bus, del tipo de

correspondencia, de la política de reemplazo, del tamaño de la caché (bloques en caché), del número de conjuntos de caché, del tamaño del bloque, del ancho de la palabra,...

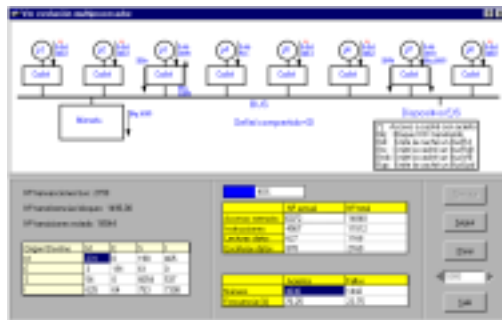


Figura 5. Visión del multiprocesador en una simulación

Además SMPCache presenta mediante datos estadísticos y todo tipo de gráficas (la figura 7 muestra un ejemplo), medidas de interés como:

- Número de transacciones en el bus.
- N° de transferencias de bloques sobre el bus.
- Tráfico en el bus teniendo en cuenta las dos medidas anteriores.
- Número de transiciones de estado (cada bloque en caché tiene un estado asociado).
- Número de transiciones de estado según el estado origen y el destino.
- Accesos a memoria, desglosándolos por tipos: captura de instrucción, lectura de dato o escritura de dato.
- Número de aciertos y fallos de caché, así como frecuencia de aciertos y fallos.



Figura 6. Diagrama de transición de estados de un bloque de caché concreto durante una simulación

Todos estos datos se presentan a distintos niveles, aunque siempre teniendo en cuenta la

interrelación de todos los elementos del sistema. P.e. podemos realizar una simulación observando el multiprocesador completo, y todos los bloques de memoria o un único bloque. También podemos observar una caché específica, y todos los bloques de memoria o un bloque concreto.

Existen tres tipos de simulación (paso a paso, con punto de interrupción y ejecución completa), admitiéndose el cambio de un tipo a otro en cualquier instante. También se puede abortar la simulación en todo momento, para corregir algún detalle de la configuración actual.

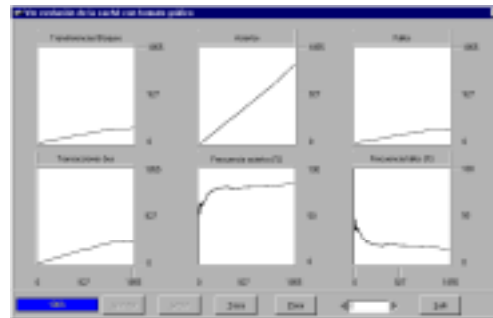


Figura 7. Datos en formato gráfico de una caché

3.2. Experiencias docentes

Podemos dividir las experiencias docentes en dos bloques. Cada uno con un grupo de guiones de prácticas.

Primer grupo de guiones de prácticas

Usando el simulador SMPCache se realizan 3 prácticas a lo largo de 3 semanas. Cada práctica tiene una finalidad concreta, con conexión directa con los aspectos abordados en la teoría:

- Ver la influencia del tamaño de la caché en la tasa de fallos y el tráfico del bus.
- Influencia del protocolo de coherencia caché sobre la tasa de fallos y el tráfico del bus.
- Influencia del número de procesadores en la tasa de fallos y el tráfico del bus.

Al igual que ocurría en la asignatura EC, las prácticas son entregadas a los alumnos en unos formularios. Estos contienen una introducción, una descripción de la arquitectura de memoria considerada, las trazas que han de utilizarse, unas tablas cuyos contenidos han de ser obtenidos con

los cálculos que efectúa el simulador, una gráfica de resultados a completar y, finalmente, una serie de cuestiones sobre la discusión de los resultados y las conclusiones obtenidas. El alumno debe realizar el experimento, rellenar el formulario y entregarlo al profesor antes de finalizar la sesión de prácticas. De esta forma se evita la copia de resultados, respuestas o conclusiones. No sólo se busca que el alumno se familiarice con el simulador, sino que también tienen gran interés las conclusiones que el alumno obtiene.

Segundo grupo de guiones de prácticas

Siguiendo este segundo grupo de guiones los alumnos deben construir una versión reducida del simulador SMPCache. Debido a la complejidad de la práctica se distinguen 4 guiones. En el primero se repasan brevemente los conceptos teóricos necesarios. Esta teoría puede ser complementada con los apuntes de las clases teóricas, así como con otras fuentes bibliográficas que le son facilitadas a los alumnos.

En el segundo se explica el entorno que el simulador debe poseer: menús, opciones relacionadas con los ficheros de configuración y las trazas de memoria, valores por defecto para la configuración del simulador,...

En el tercero se analiza en profundidad la implementación software de todos los conceptos de teoría de cachés, a tener en cuenta para desarrollar el simulador. Se muestra, mediante pseudocódigo, una posible solución para la implementación software de los protocolos de coherencia caché, las funciones de correspondencia, las políticas de reemplazo y los esquemas de arbitraje del bus.

En el cuarto, y último, se explican detalladamente los datos a mostrar en pantalla durante los distintos tipos de simulación: texto y gráficos, paso a paso, con punto de interrupción y ejecución completa, etc.

El segundo guión de prácticas se complementa con las explicaciones del profesor durante las clases prácticas. Los alumnos también aprovechan estas clases para ir construyendo su propio simulador, cuya bondad pueden comprobar comparando sus resultados con los obtenidos con SMPCache. La construcción del simulador obliga a los estudiantes a conocer en profundidad las consideraciones teóricas sobre memorias caché, y

particularmente dentro de multiprocesadores (coherencia caché, protocolos de coherencia,...).

Tras finalizar el periodo de prácticas los alumnos, en grupos de dos, entregan en disco la versión definitiva del simulador realizado: fichero ejecutable y fuentes. También entregan un documento escrito donde aparezcan los siguientes apartados: listado comentado del programa, manual de programador y manual de usuario.

Para asegurar la autoría de las prácticas y que éstas cumplen unos requisitos mínimos de calidad, todos los alumnos pasan por un examen de defensa individual. A los alumnos, que aprueban este examen, se les evalúa el simulador y documentación entregada.

4. Conclusión

El simulador SISMEC se ha utilizado durante los últimos 5 años; mientras SMPCache se lleva usando los 3 últimos años. Nuestras experiencias, en este largo periodo de uso, nos han demostrado los beneficios de ambos simuladores con propósitos educacionales. Ambos han tenido una buena aceptación por parte de los alumnos. Además, hemos observado que con su uso los estudiantes adquieren un conocimiento mejor y más profundo sobre memorias cachés. Ambos simuladores están disponibles de forma gratuita, con fines docentes y de investigación [2].

Referencias

- [1] Culler, D.E.; Singh, J.P.; Gupta, A. *Parallel Computer Architecture. A Hardware/Software Approach*. Morgan Kaufmann, 1999.
- [2] GACDL. <http://atc.unex.es/gacdl>, 2001.
- [3] Hennesy, J.L.; Patterson, D.A. *Computer Architecture - A Quantitative Approach*. Morgan Kaufmann, 1996.
- [4] Patterson, D.A.; Hennesy, J.L. *Computer Organization and Design. The Hardware/Software Interface*. Morgan Kaufmann, 1994.
- [5] Sima, D.; Fountain, T.; Kacsuk, P. *Advanced Computer Architectures. A Design Space Approach*. Addison-Wesley, 1998.