

Diseño de experiencias prácticas sobre memoria cache

X. Molero, Á. Rodas, A. Pont, J. Sahuquillo, J.M. Valiente

Escola Universitaria d'Informàtica
Departament d'Informàtica de Sistemes i Computadors
Universitat Politècnica de València
Camí de Vera, 14. 46022 València
e-mail: jmolero@disca.upv.es

Resumen

La planificación de actividades prácticas a fin de consolidar el aprendizaje de los conceptos clave relacionados con la memoria cache no es una tarea fácil.

En esta ponencia destacamos los conceptos que hemos considerado fundamentales sobre el sistema de memoria cache. A continuación planteamos un conjunto de actividades prácticas basándonos en el refuerzo de dichos conceptos y establecemos la metodología para su enseñanza. Finalmente, presentamos una herramienta de ayuda, accesible mediante un navegador web, que apoyará el proceso de aprendizaje.

1. Introducción y motivación

El concepto de jerarquía de memoria, y dentro de él, el sistema de memoria cache, es un elemento fundamental en cualquier curso introductorio sobre estructura o arquitectura de computadores [4,5,7]. En el caso que nos ocupa, la docencia de este tema tiene lugar en la asignatura Estructura de Computadores II. Esta asignatura es obligatoria y se sitúa en el tercer cuatrimestre de los planes de estudio de las titulaciones de ingeniería informática, tanto superior como técnicas, que se imparte en la Escuela Universitaria y en la Facultad de Informática de la Universitat Politècnica de València.

Dada la relevancia que actualmente tiene el sistema de memoria cache dentro de los computadores actuales, la localización de los conceptos teóricos clave es una tarea fundamental, así como su refuerzo mediante una buena planificación de las actividades prácticas. En esta línea, los autores se han propuesto una doble tarea: aislar aquellos aspectos relevantes en la organización del sistema

de memoria cache, y proponer una serie de experiencias prácticas que ayuden al alumno a una óptima comprensión de los conceptos clave. Por último, hemos creído necesario apoyar este aprendizaje práctico sobre una herramienta diseñada con una finalidad exclusivamente didáctica, y al mismo tiempo que sea fácil de usar para la consecución de las actividades prácticas.

El resto de esta ponencia se ha organizado de la siguiente manera: la Sección 2 presenta los conceptos más destacables relacionados con el sistema de memoria cache. En la Sección 3 se presenta un conjunto de actividades prácticas diseñadas en torno a los conceptos fundamentales. La metodología que proponemos de aprendizaje se describe en la Sección 4, mientras que la herramienta que proponemos para utilizarse en las actividades prácticas se describe brevemente en la Sección 5, así como un ejemplo de utilización. Por último, las conclusiones se detallan en la última sección.

2. Identificación de conceptos clave

Las experiencias prácticas deben plantearse de manera que incidan en los conceptos clave que, a nuestro entender, aparecen cuando nos enfrentamos con la enseñanza de la memoria cache. No debemos olvidar que ésta es la primera vez dentro de los estudios universitarios en que al alumno se le presenta la memoria cache. En consecuencia, nuestro enfoque ha de ir necesariamente referido a la parte más descriptiva, dejando los aspectos relacionados con el rendimiento para asignaturas posteriores. En esta sección justificamos brevemente la elección de los elementos de la memoria cache que hemos creído más importantes, y que serán los reforzados en las sesiones prácticas de la asignatura.

La Figura 1 muestra gráficamente el marco de referencia que los autores han utilizado para construir la base de las experiencias prácticas. Este transfondo se ha dividido en dos partes. La primera establece los parámetros más importantes que definen la configuración de la memoria cache. La segunda plantea los dos índices de comportamiento utilizados para medir la bondad de la memoria cache cuando se ve sometida a una serie de accesos (conjunto de referencias a memoria emitidas por el procesador).

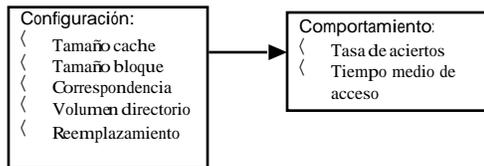


Figura 1. Conceptos fundamentales a considerar sobre memoria cache

2.1. Parámetros de configuración

La justificación histórica de la memoria cache es la misma que explica por qué funciona la jerarquía de memoria del computador: el principio de localidad. Si éste no se cumpliera en la práctica, toda la jerarquía de memoria carecería de sentido. Así pues, la memoria cache no es más que un nivel, o conjunto de niveles, entre los registros del procesador y la memoria principal. Planteado así, el concepto siguiente es una consecución lógica del planteamiento anterior, e ilustra el hecho de clasificar los accesos a memoria cache en aciertos y fallos. Si el acceso es un acierto, la información puede fluir hacia el procesador sin necesidad de acceder a memoria principal; en caso contrario, el acceso se cataloga como fallo. La ocurrencia de un fallo desencadena una serie de acciones encaminadas a proporcionar la información accediendo a la memoria principal y, en general, actualizando el contenido de la memoria cache. Tras las consideraciones anteriores es necesario establecer las unidades de transferencia entre los niveles de la jerarquía compuestos por el procesador, la memoria cache y la memoria principal. En este contexto se define el concepto de bloque (o línea) como la unidad de transferencia entre la memoria cache y la memoria principal.

Por otro lado, si se comparan las capacidades de almacenamiento de la memoria cache y la memoria principal, se pone de manifiesto la necesidad de establecer algún criterio para ubicar los bloques de memoria principal en memoria cache. De esta manera surgen las diferentes correspondencias: directa, totalmente asociativa y asociativa por conjuntos. La necesidad de esta ubicación determina el uso de etiquetas para averiguar si el bloque referenciado se encuentra en memoria cache, y por tanto, aparece también el concepto de directorio. Además, será necesario saber si la información contenida en un bloque de memoria cache es válida, y por tanto, hará falta un bit de estado (bit de válido).

Al hilo del razonamiento anterior, se hace patente la necesidad de establecer alguna política de reemplazo de los bloques de la memoria cache, como la menos recientemente usado (LRU, *least recently used*), orden de llegada (*first in, first out*), aleatoria (*random*), etcétera. Estas políticas necesitarán de bits adicionales para su gestión.

En cuanto a los accesos a memoria, su clasificación en lecturas y escrituras permite presentar el problema que acontece en el segundo caso, ya que hay que decidir si se escribe en ambas memorias (política de actualización directa, *write through*), o bien se escribe sólo en la memoria cache y posteriormente se actualiza la memoria principal (actualización posterior, *write back*). Esta última política necesitará un bit de estado adicional (bit de modificado) que indique si un bloque hay que actualizarlo o no en memoria principal en el momento en que se reemplace.

2.2. Análisis de comportamiento

El comportamiento dinámico de memoria cache depende de su configuración y del patrón de accesos que realice el procesador. Tradicionalmente este comportamiento se puede analizar fácilmente mediante dos variables: la tasa de aciertos y el tiempo medio de acceso.

La tasa de aciertos (H) obtenida tras realizar una serie de accesos a memoria cache refleja la proporción de accesos que han sido aciertos, y por tanto, se calcula dividiendo el número de aciertos entre el número total de accesos. Este valor depende principalmente del principio de localidad de las referencias, así como de la correspondencia y de otros parámetros de configuración, y por tanto

incide directamente en el tiempo medio de acceso que experimenta el procesador.

El tiempo medio de acceso al sistema de memoria se puede calcular analíticamente mediante la fórmula $H \times t_a + (1-H) \times t_f$, donde t_a es el tiempo de acceso en caso de acierto (tiempo de acceso de la memoria cache), y t_f es el tiempo necesario para resolver el fallo, tiempo este último que depende del tamaño de bloque y de la organización de la memoria principal. En cualquier caso, un buen diseño del sistema de memoria intentará reducir el valor de t_f , y en este sentido la memoria principal ha de diseñarse para optimizar el acceso a bloques: organizando la memoria en módulos, permitiendo el acceso optimizado a ráfagas de datos, o bien utilizando memorias de gran ancho de palabra.

3. Propuesta de actividades

Una vez identificados los conceptos clave que el alumno debe conocer y comprender sobre la memoria cache, se hace necesario plantear un conjunto de actividades que le permitan la adquisición de dichos conceptos. Un primer grupo de actividades se dedicará al estudio de la configuración de la memoria cache y un segundo grupo tratará aspectos relacionados con el análisis de su comportamiento ante diferentes supuestos.

3.1. Actividades sobre configuración

Este primer grupo de actividades tiene por objetivo la adquisición de conocimientos básicos relacionados con la información de control presente en la memoria cache y la forma en que ésta se maneja para distintos supuestos.

En algunos casos, puede ser un objetivo indirecto de dichas actividades la adquisición de destrezas simples por parte de los alumnos (cálculo de direcciones totales, bits de desplazamiento, etc.).

Como ejemplo de actividades sobre configuración proponemos las siguientes:

- Distribución de los bits de dirección desde el punto de vista de la memoria cache.
- Cálculo del tamaño del directorio.
- Descripción de la organización interna: bits de estado, comparadores, contadores, etc.

En todos los casos se pueden establecer comparaciones de los resultados obtenidos para distintos supuestos de configuración.

3.2. Actividades de análisis de supuestos

En este caso, el objetivo que se persigue es el desarrollo de destrezas complejas y de habilidades para la toma de decisiones. Se trata pues de actividades más elaboradas que generalmente deben ir precedidas de actividades sobre configuración.

Se pretende que el alumno estudie el efecto que produce el cambio de algún o algunos de los parámetros de la memoria cache, así como el análisis de distintos patrones de acceso sobre una o varias configuraciones.

Proponemos como actividades dentro de este grupo el estudio del efecto sobre las variables de rendimiento en los siguientes casos:

- Según el grado de localidad de los patrones de acceso.
- Según el tipo de correspondencia.
- Según la política de reemplazo.
- Según la política de escritura.
- Según la organización de memoria principal.

En todos los casos se puede optar por un análisis individual más sencillo o por análisis comparativos que puedan suponer una toma de decisiones por parte del alumnos. En ese sentido resulta apropiado presentar, además de casos típicos, casos extremos donde se potencie o se inhiba las ventajas de cada configuración.

Por ejemplo, en la primera actividad que se refiere al análisis del rendimiento según localidad de los patrones, se trataría de presentar al alumno tres tipos de patrones de referencia:

- Patrones que potencien la bondad de la configuración de la cache (ejemplo: referencias consecutivas en cualquier correspondencia).
- Patrones que inhiban la bondad de la configuración (ejemplo: referencias a bloques de memoria principal que van al mismo bloque de memoria cache, cuando se trata de una correspondencia directa)
- Patrones intermedios.

4. Metodología

Una vez definidos los objetivos y actividades necesarios para el aprendizaje de la memoria cache, deberemos optar por una propuesta metodológica adecuada a los mismos.

Sin embargo, deberemos considerar, además, un conjunto de circunstancias que actualmente conforman el panorama de la enseñanza universitaria:

- Incremento del número de alumnos que comparten el estudio con el trabajo.
- Gran oferta de asignaturas optativas con el problema de dificultad de ajustar horarios.
- Aumento de la carga lectiva en prácticas.

Todo ello unido a la proliferación de Internet, hace que se tienda a un sistema de enseñanza no presencial o semipresencial que está dando buenos resultados en muchas universidades [2].

Con todo ello, en la enseñanza práctica de la memoria cache, optamos por una propuesta metodológica basada en el autoaprendizaje mediante métodos multimedia y acceso a través de internet [1]. En ese sentido nos parece adecuado el uso de simuladores que ayudan al desarrollo de destrezas complejas y habilidades para la toma de decisiones. Dichos simuladores pueden acompañarse de programas tutoriales, para ayudar a adquirir conceptos básicos, o incluso acompañarse de programas de ejercitación para desarrollar destrezas simples en aquellos casos en que se precisen.

Por otra parte, hay que tener presente que la asignatura Estructura de Computadores establece los conceptos fundamentales en que se basa el sistema de memoria cache. Es por ello que se ha evitado plantear su estudio en términos de rendimiento, materia que es tratada en asignaturas de cursos posteriores.

5. *WebCache*: un ejemplo de herramienta de trabajo

La elección de una herramienta para llevar a cabo la parte práctica está basada en las facilidades que ofrece para cumplir los requisitos establecidos en las Secciones 3 y 4. En este sentido, el programa informático que proponemos para su uso por parte de los alumnos ha sido diseñado específicamente con un propósito docente [3].

A continuación pasamos a exponer las características de dicho programa ilustrando su funcionamiento con un ejemplo sencillo.

5.1 Características principales de la aplicación

Esta herramienta se ha desarrollado en lenguaje Java, y se ejecuta como una *applet* a través de Internet. Así pues, para utilizarla sólo es necesario disponer de un programa navegador de Internet y un entorno de ventanas. En cualquier caso, el programa también puede grabarse en un disquete y ejecutarse de manera local en el computador.

La aplicación es accesible, por tanto, desde cualquier parte del planeta a través de Internet y puede utilizarse como herramienta educativa en un curso introductorio sobre estructura de computadores. Además, y dada la casuística de los alumnos matriculados en nuestra universidad, el programa puede usarse en tres idiomas diferentes: castellano, catalán e inglés.

La parte más novedosa es la capacidad de simular gráficamente la evolución del estado de aquellos elementos que forman parte de un sistema de memoria cache. Se puede simular cada acceso a memoria de forma individual, siempre bajo el control del usuario, y observar, por ejemplo, el contenido de las líneas, bits de estado, contadores de los algoritmos de reemplazo, tiempo de acceso transcurrido, número de fallos, tasa de aciertos, etcétera.

El programa se encuentra disponible para su uso en la dirección URL especificada en [6].

5.2. Ejemplo sencillo de ejecución

El menú reflejado en la Figura 2 resume los parámetros más importantes que pueden indicarse del sistema de memoria:

- Memoria cache: capacidad de la cache (expresada en KB y potencias enteras de 2), tamaño del bloque y tiempo de acceso.
- Memoria principal: capacidad total (expresada en KB), hasta un máximo de 1 GB y tiempo de acceso.
- Tipo de correspondencia: directa, totalmente asociativa y asociativa por conjuntos, en la que también habrá que elegir el grado de asociatividad.

- Organización de la memoria principal: secuencial, palabra ancha y entrelazada (organizada en módulos).
- Algoritmo de reemplazo de bloques en memoria cache: menos recientemente usado, orden de llegada y aleatorio.
- Modo de escritura: actualización directa y actualización posterior.
- Simulación gráfica: el simulador ofrecerá la posibilidad de seleccionarla cuando la configuración elegida, tanto de memoria cache como de memoria principal, pueda ser representada gráficamente en la pantalla. En este caso, se pueden simular gráficamente todas aquellas configuraciones con 16 o menos bloques en memoria cache.

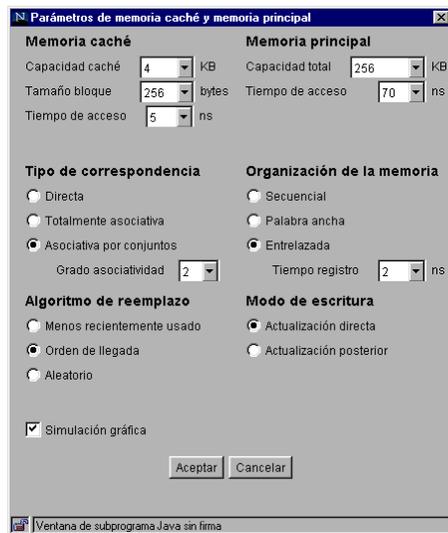


Figura 2. Configuración del sistema de memoria

El patrón de las referencias se puede indicar de tres maneras. La primera da la posibilidad de introducir las referencias en formato de acceso simple (se puede elegir la base numérica en que se especifican las referencias: hexadecimal o decimal). Por ejemplo, la cadena $10, r, 2-1023, w;$ indica una lectura de la posición 10, y escrituras en las posiciones consecutivas desde la 2 hasta la 1023. La segunda especifica un bucle simple, y permite iterar una secuencia de accesos contiguos un número determinado de veces (lecturas o escrituras). La tercera es un bucle compuesto

y proporciona la posibilidad de poder anidar dos bucles simples.

Supongamos un conjunto de referencias que viene dado por accesos de lectura desde la posición 0 hasta la 127, ambas inclusive, un total de tres veces consecutivas. La especificación de estas referencias en el programa se muestra en la Figura 3. Esta ventana permite también reiniciar el estado de la memoria cache y así probar diferentes referencias para el mismo sistema de memoria que se encuentre definido en ese momento.



Figura 3. Especificación del patrón de referencias

Se ha supuesto una memoria principal de 32 KB y una memoria cache de 1 KB. El bloque tiene un tamaño de 64 bytes. La correspondencia es totalmente asociativa y el algoritmo de reemplazo empleado es el menos recientemente usado (LRU).

Por otra parte, el tiempo de acceso a memoria cache se ha considerado de 10 ns, y el de memoria principal de 50 ns. La memoria principal es entrelazada, con un tiempo de acceso al registro de datos de cada módulo de 2 ns. En la Figura 4 se muestra el resultado final de la simulación del patrón de accesos anterior.

Aunque aquí se ha mostrado el resultado final, el alumno puede comprobar el efecto de cada acceso por separado, y así mismo, cómo van evolucionando la tasa de aciertos y el tiempo medio de acceso.

Por último, esta aplicación dispone de un tutorial sobre el concepto y funcionamiento de la memoria cache que puede ser útil para revisar conceptos conforme se vayan realizando las simulaciones. En su diseño se persigue que el alumno pueda hacer uso del hipertexto, para desplazarse de una manera cómoda a lo largo del tutorial.

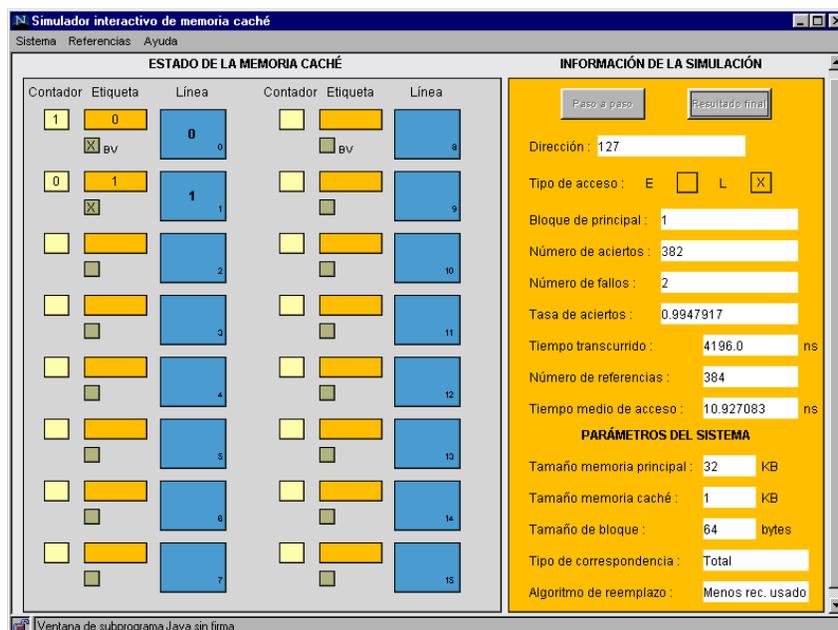


Figura 4. Estado final de la memoria caché

6. Conclusiones

La enseñanza de la memoria cache debe complementarse con un conjunto de actividades prácticas que sirvan para reforzar los conceptos clave. Con este objetivo, la presente ponencia identifica estos conceptos y plantea un modelo concreto de actividades.

Por otro lado, y teniendo en cuenta las circunstancias actuales que rodean la enseñanza universitaria, hemos propuesto una metodología de aprendizaje acorde a este entorno.

La materialización de nuestra propuesta cuenta con el apoyo de una herramienta de trabajo que se adecúa a nuestras necesidades: entorno visual, facilidad de uso, ejecución desde internet, multilingüe, etc.

Referencias

- [1] A.R. Bartolomé, *Sistemas Multimedia*. Sancho, J. (coord.) *Para una Tecnología Educativa*. Barcelona: Horsori. pp. 193-219. 1994.
- [2] A.R. Bartolomé, *Save the University. About Technology and Higher Education*. (colab. con Dr. Lauran Sandals). Educational Multimedia and Hypermedia annual, 1998. AACE: Charlottesville. Ottman e I. Tomek (Ed.) (VA). pp. 111-117.1998
- [3] X. Molero, J.C. Campelo, F. Rodríguez, V. Santonja, J.J. Serrano. *WebCache: a graphic simulation tool of cache memories using the internet*. International Conference on Simulation and Multimedia in Engineering Education, 2000.
- [4] A. Pont, J.A. Gil, M. Martínez, X. Molero, A. Robles, Á. Rodas. *Estructura de computadores (volumen 2). De la memoria del computador a la entrada/salida*. Servicio de publicaciones de la Universidad Politécnica de Valencia, 1998.
- [5] D.A. Patterson, J.L. Hennessy. *Estructura y diseño de computadores. Volumen 2*. Reverté, 2000.
- [6] Simulador de memoria cache. Disponible en <http://www.disca.upv.es/jmolero/ec2/index.htm>
- [7] W. Stallings. *Organización y arquitectura de computadores*. Prentice Hall, 2000.