

CACHESIM: UN SIMULADOR DE CACHE CONFIGURABLE

M^a Luisa Córdoba Cabeza¹, M^a Isabel García Clemente¹, M^a Luz Rubio Martínez¹

¹*Departamento de Arquitectura y Tecnología de Sistemas Informáticos (DATSI)
Facultad de Informática (UPM)
e-mail: mcordoba@fi.upm.es*

RESUMEN: Los conceptos de memoria virtual y jerarquía de memoria son importantes en la docencia de arquitectura de computadores. La mayoría de los libros de texto presentan estos temas como una descripción completa de conceptos básicos, modos de funcionamiento, y algoritmos. Pero es básico completar esto con ejemplos prácticos de diseño de caches reales, en la que puedan observar de forma empírica la influencia de las decisiones de diseño sobre la jerarquía de memoria en el rendimiento global del sistema.

1.- INTRODUCCIÓN.

Este trabajo presenta un simulador de jerarquía de memoria de un computador (CacheSim) desarrollado en el D.A.T.S.I. de la Facultad de Informática de la UPM. Este simulador está orientado a alumnos de Arquitectura de Computadores. Pero la herramienta se puede utilizar tanto para el desarrollo de prácticas en laboratorio, como para el aprendizaje por computador de la teoría de este tema, gracias a un completo sistema interactivo de ayuda.

El simulador se utilizaría para realizar prácticas en las que los alumnos diseñen su propia jerarquía de memoria. Dado un programa fuente (representativo de una determinada carga de trabajo típica), los alumnos deberían seleccionar de forma óptima los parámetros que caracterizan la jerarquía de memoria.

Los alumnos también pueden comprobar y cuantificar cómo influyen los distintos valores de algunos parámetros de la jerarquía en las prestaciones y rendimiento del sistema de memoria.

2.- OBJETIVOS.

El principal objetivo en el desarrollo de esta herramienta era construir no sólo una herramienta pedagógica, sino también innovadora.

Se ha desarrollado un simulador del funcionamiento de la jerarquía de memoria de un computador centrándose en la memoria cache y en la memoria principal. Calcula tasas de fallos, tasas de aciertos, tiempos de ciclo de memoria, tiempos medios de acceso a memoria para una configuración de jerarquía de memoria determinada.

Las características principales de la herramienta son:

1. Pedagógica
2. Accesible
3. Fácil de utilizar
4. Poco esfuerzo del alumno
5. Innovadora
6. Modular

El simulador se ejecuta en PC, bajo Windows. Esto permite facilidad de acceso a la aplicación. Es un programa Windows, por lo que el interfaz y entorno de ejecución es muy extendido y conocido por los alumnos.

La interfaz con el usuario se ha diseñado para facilitar tanto la ejecución de la simulación, como la introducción de los valores de los parámetros de configuración.

Otra característica es que CacheSim proporciona un sistema de ayuda sensible al contexto, en el que cabe resaltar:

- Incluye toda la teoría sobre las jerarquías de memoria. Los alumnos pueden simultanear su consulta en cualquier instante mientras realizan la práctica. Se pueden utilizar diversos tipos de búsqueda por índice, y acceder a un libro de texto *on line* rápido mientras se realiza la práctica. Esto permite al alumno que sea capaz de comenzar una práctica sin un conocimiento profundo previo sobre memorias cache, guiándole sobre los principales conceptos, el comportamiento, y posibles mejoras. El aprendizaje de la teoría es una tarea menos tediosa.
- Dirige al alumno durante la ejecución del simulador y le enseña a utilizarlo, de modo que es autoexplicativa.
- Detección de errores en la asignación de valores: parámetros de la configuración de la jerarquía de memoria, y variables de ejecución de la simulación. Por ejemplo, la introducción de parámetros que tienen opciones limitadas, es a través de gráficos en forma de listas en cascada.

Todo esto facilita la introducción de los datos de entrada, la ejecución de la simulación, con un esfuerzo mínimo de comprensión del interfaz y la operatividad de la herramienta. Todas estas características se resumen en:

- El alumno se puede concentrar su esfuerzo en el estudio del diseño y optimización de la jerarquía de memoria
- Utilización sencilla y cómoda de la herramienta. Se facilita la introducción de variables y la repetición de experimentos.
- La obtención de los datos de entrada y la visualización de los resultados sean tareas sencillas. Se enlaza con WgnuPlot, para visualizar gráficamente los resultados de rendimiento obtenidos en distintas simulaciones.

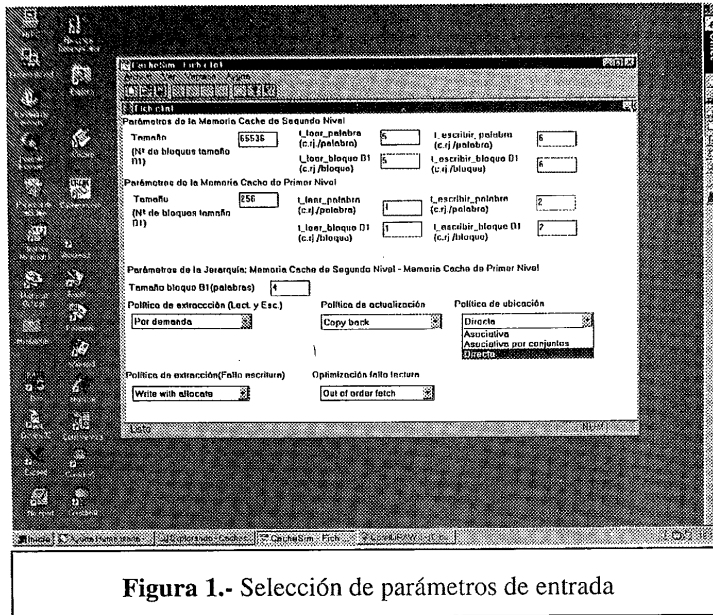


Figura 1.- Selección de parámetros de entrada

CacheSim es además innovador respecto a otras herramientas similares (DineroIII, SISMEC), ya que incluye :

- Calcula tiempos medios de acceso y tiempos de ciclo de memoria para tiempos de acceso tanto de palabra como de bloques, seleccionados por el usuario, en cualquier nivel de la jerarquía.
- Permite elegir tiempos de acceso para palabras y para bloques. Esto hace que sea posible mejorar el ancho de banda entre dos niveles adyacentes de la jerarquía de memoria. Al tener que proporcionar tiempos para palabras y para bloques, se puede aumentar el tamaño del bloque que se intercambia entre dos niveles de la jerarquía sin el correspondiente incremento en la penalización de fallos. El número de arquitecturas que es posible simular es mayor que con otras herramientas.
- Implementa la política de reemplazo generada por la aplicación del algoritmo LRU*. Este algoritmo se utiliza en las memorias cache multinivel cuando el segundo nivel de memoria cache tiene una política de ubicación asociativa o asociativa por conjuntos.

3.- EL SIMULADOR CACHESIM

CacheSim es un simulador de jerarquía de memoria cuyo sistema de memoria está compuesto por uno o dos niveles de memoria cache y un nivel de memoria principal. Es un simulador dirigido por traza. La entrada de datos es un fichero de traza con las referencias a memoria de una ejecución de un programa. Antes de la simulación hay que generar esta traza. Ésta se puede generar manualmente, o con otros programas de libre difusión como DLXsim.

CacheSim simula tres configuraciones posibles:

- Memoria cache única
- Memoria cache de datos separada de la memoria cache de instrucciones
- Dos niveles posibles de cache

Antes de la simulación también hay que configurar la jerarquía de memoria: tamaños y tiempos de acceso de caches, tamaño y tiempo de acceso de memoria principal, relación entre cache y memoria principal (tamaño de bloque), y todas las políticas de intercambio de bloques.

Durante la simulación se supone la propiedad de inclusión multinivel: todos los datos contenidos en la memoria de un nivel están incluidos también en la del nivel siguiente. Por lo tanto, en un fallo de cache de segundo nivel hay que invalidar todos los bloques del primer nivel correspondientes. Esto incrementa ligeramente la frecuencia de fallos del primer nivel.

La configuración de la memoria y las políticas se definen mediante ficheros. Diferentes tipos de ficheros contienen los parámetros de entrada. Se pueden crear ficheros de entrada totalmente nuevos, o modificando alguno existente.

Los resultados de la simulación (tasas de aciertos, tasas de fallos, tiempos medios de accesos, tiempos medios de ocupación) también se obtienen en ficheros.

Como salida opcional también se puede obtener un fichero de volcado con traza de la simulación con los contenidos de memoria y resultados tras cada acceso a memoria: cuál es el número de bloque de memoria principal de la dirección de memoria procesada, dónde están los bloques de memoria principal, los valores de los bits de validez y ensuciado, contenido de la lista del algoritmo de reemplazo, etc... Los alumnos pueden estudiar este fichero de volcado para diferenciar los distintos tipos de fallos de cache:

- fallos forzosos
- fallos por capacidad
- fallos por conflicto

La herramienta enlaza con WgnuPlot para directamente, si se desea, representar gráficamente los resultados de una o varias simulaciones.

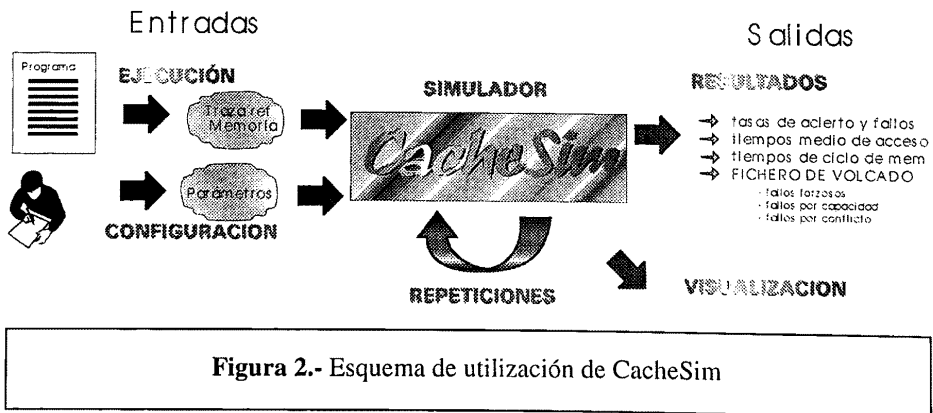


Figura 2.- Esquema de utilización de CacheSim

4.- EJEMPLOS

A modo de ejemplo de una práctica, mostramos los resultados que los alumnos obtendrían tras simular repetidamente una traza, variando algunos parámetros de configuración mientras otros permanecen constantes. En estos ejemplos nos hemos centrado en estudiar las variaciones de las tasas de aciertos, aunque también observamos otros resultados indicadores de prestaciones (tiempo de acceso, tiempo de ocupación o de ciclo, tasa de fallos, etc). Los resultados se han representado con WgnuPlot.

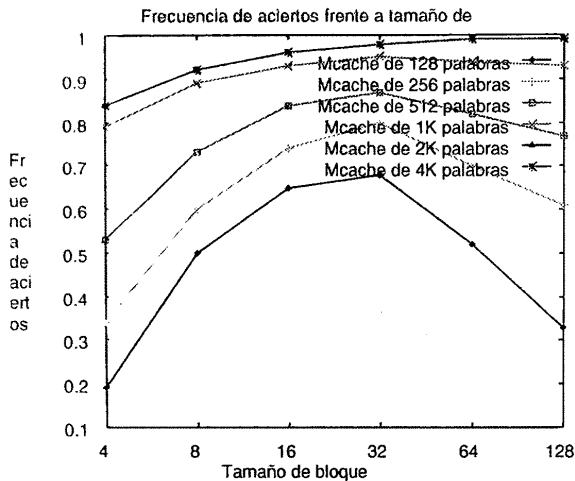


Figura 3.- Tasa de aciertos para caché única y distintos tamaños de caché y bloque

Cache única. Configuración: Memoria principal: tamaño 1M pal; tiempo de acceso a palabra 10 c; tiempo de acceso a bloque 10 c; política de ubicación directa.

Fetch por demanda; política de escritura *copy back*; política de escritura en fallo *write with allocate*; política de lectura *out of order fetch*

Memoria caché: tiempo de lectura de palabra 1c; tiempo de escritura de palabra 2c; tiempo de lectura de bloque 1c; tiempo de escritura de bloque 2c.

5.- CONCLUSIONES

En este artículo se presenta un simulador de jerarquía de memoria para propósitos didácticos: CacheSim. Es una herramienta pedagógica, fácil de utilizar, y con un sistema de ayuda sensible al contexto que además de guiar al alumno a través de la simulación, le explica todos los conceptos teóricos de jerarquía de memoria. Su utilización sencilla permite al alumno centrar sus esfuerzos en los conceptos de jerarquía de memoria en vez de aprender el manejo de la herramienta. Este simulador enseña al alumno además cómo diseñar una jerarquía de memoria, y le permite observar y cuantificar las consecuencias en las prestaciones y rendimiento del sistema debidas a sus elecciones en el diseño.

En cuanto a líneas de continuación del trabajo, tenemos planeado modificar los formatos de los datos de entrada. Además de un simulador dirigido por traza, estamos tratando de conectar el simulador con un generador de trazas de libre difusión para que la entrada al simulador pueda ser un programa en código fuente. Los alumnos podrían estudiar cómo programar un algoritmo de forma que se obtenga el mejor rendimiento de la configuración de la jerarquía de memoria. Esto les permitiría identificar en qué instrucciones del programa se producen los fallos. Sería posible entonces, trabajar en técnicas de optimización de compiladores tales como el intercambio de bucle y la fusión de bucle o bloqueo.

6.- BIBLIOGRAFÍA

- [1]. University of California.
Dlxc-DLX Compiler (Dlx Specific Implementation of gcc)
University of California, 1988.
- [2]. García, M. Isabel.
Sistema de Memoria.
Publicaciones, Univ. Politécnica de Madrid, 1998.
- [3]. Córdoba, M.L.; García, M.I.; Rubio, M.L.
CacheSim: A Cache Simulator for Teaching Memory Hierarchy Behaviour.
Proc. Of 4th Annual SIGCSE/SIGCUE (ACM). ITiCSE'99. Krakow, Poland, 1999.
- [4]. Gómez, J.A.; Sánchez, J.M.; Moreno, J.A.
An Educational Tool for testing Hierarchical Multilevel Caches.
Dept. Informática de la Universidad de Extremadura, 1997. Hostetler, L.B.; Mirtich, B.
DLXSim - A Simulator for DLX.
November, 1993. Wang, Wen-Hann.
Multilevel Cache Hierarchies.
Tech. Report 89-09-13. Dept. of Comp. Science and Engineering. FR-35. Univ. of Washington,
1989.
- [5]. Kotz, D.
Latex and the Gnuplot Plotting Program (ver. 3.0).
david.kotz@dartmouth.edu July, 1991.
- [6]. David A. Patterson, John L. Hennessy.
Computer Organization and design. The hardware/software interface. Second Edition.
Morgan Kaufmann Publishers, Los Altos, USA, 1998.