Arquitectura de Computadores

Práctica 1

Jerarquía de memoria y comportamiento de memoria caché: Estudio del efecto de la localidad de los accesos a memoria en las prestaciones de programas en microprocesadores

Objetivos: mediante la realización de un programa en lenguaje C, caracterizar el coste temporal por lectura (acceso) de datos, variando diferentes parámetros en el patrón de acceso y el tamaño del conjunto de datos, y observar e interpretar el efecto del sistema de memorias cache del microprocesador teniendo en cuenta los conceptos de localidad espacial, localidad temporal y precarga (*prefetching*).

Equipos: grupos de prácticas de dos personas.

Sistema para la toma de datos: ordenadores del aula de informática o propios.

Tiempo de trabajo presencial: 4 sesiones

Plazo de entrega: hasta el viernes de la semana siguiente a la semana en la que tenga lugar la última sesión asociada a esta práctica.

Forma de entrega y formato: Electrónico. Se creará una tarea en el campus virtual para entregarlo. Es necesario entregar listados de los códigos además del informe de prácticas correspondiente.

Valoración: 44% de la nota de prácticas (hasta 2.2 puntos sobre 10 en la nota total).

Hacer un programa en C que reserve memoria dinámica un vector (*array*) A[] de números representados en punto flotante tipo *Double*. Llevar a cabo una operación de reducción de suma de punto flotante sobre los R elementos del vector siguientes: A[0], A[D], A[2D], A[3D],....,A[(R-1)*D] (es decir, sumar estos R elementos almacenando el valor en una variable tipo *Double*). D es un parámetro que vamos a variar para estudiar el efecto de la localidad. La referencia a los elementos de A[] debe hacerse indirectamente a través de un vector de enteros ind[], es decir, hacer referencias del tipo A[ind[i]], con ind[i]=0, D, 2D, 3D, etc.

Repetir 10 veces la ejecución de la reducción para luego calcular el valor medio. Para cada repetición almacenar el resultado en un vector S[] de 10 elementos tipo *Double*. Imprimir todos los elementos de S[] (los resultados) al terminar. Mediante este proceso evitamos que el compilador realice optimizaciones que en este caso no deseamos.

Tomar la medida de ciclos (utilizando las rutinas que aparecen en el programa C adjunto) totales de las 10 repeticiones (no incluir la impresión de los resultados en esta medida), y obtener el **número de ciclos medio por acceso a memoria**.

Repetir el experimento para diferentes valores de R y D.

Valores de R: para un valor específico de D, es necesario realizar un total de 7 medidas. En cada medida seleccionamos el valor de R de modo que las lecturas del vector A[] correspondan a un total de L líneas caché diferentes. Tomar medidas de ciclos para los siguientes valores de L: 0.5*S1, 1.5*S1, 0.5*S2, 0.75*S2, 2*S2, 4*S2, 8*S2, siendo S1 el número de líneas caché que caben en la caché L1 de datos y S2 el número de líneas caché que caben en la caché L2.

Será necesario investigar cuales son los valores de S1 e S2. Es necesario estar seguro de que estos valores son los correctos, ya que determinarán la calidad de los experimentos.

Valores de D: 5 valores de D entre 1 e 100. Los valores de D deben seleccionarse de modo que los experimentos realizados permitan estudiar de forma representativa los diferentes efectos de la localidad de acceso a los datos. Por ejemplo, no tendría sentido utilizar D=90, 91, 92, 93, 94 e 95, ya que es probable que los resultados sean muy similares y, por lo tanto, no permitirían interpretar otros efectos de la localidad que puedan ser de interés. La selección de los valores de D adecuados es una parte muy importante del experimento.

Número de elementos do vector A[]: determinado por los valores seleccionados de R y D. Para la reserva de memoria resulta conveniente alinear el comienzo del vector con el comienzo de una línea cache. Una alternativa para hacer esto es utilizar la función _mm_malloc(TC,CLS), donde TC es el número de bytes a reservar y CLS es el tamaño de línea cache en bytes. Esta función devuelve un puntero alineado a CLS bytes. Para liberar la memoria se recomienda utilizar la correspondiente función _mm_free(P), donde P es el puntero que apunta al comienzo del vector de memoria reservado. Para utilizar estas funciones es necesario poner esta línea en el archivo: #include <pmmintrin.h>. En la compilación con gcc es necesario utilizar la opción -msse3.

Se proporciona con este enunciado un programa C con las rutinas de medida de tiempo necesarias que incluye también la rutina mhz(). Esta rutina imprime la frecuencia de reloj del procesador.

Presentación de resultados: Hacer una representación gráfica del coste en ciclos de cada acceso a memoria (eje Y) vs. el número total de líneas cache diferentes del vector A[] que se referencian (eje X) para los diferentes valores de D. Se pueden hacer representaciones gráficas adicionales si se considera necesario para explicar los resultados e incorporar tablas de datos.

Interpretación de resultados: El informe debe incluir una interpretación de los resultados obtenidos en relación a los conceptos de localidad temporal y espacial en el acceso a los datos. En los microprocesadores se suele realizar precarga (*prefetching*) de datos, busca información sobre cómo se realiza la precarga en el procesador que utilizas.

Variabilidad de los resultados: es recomendable tomar varias medidas para un mismo experimento para observar la variabilidad. Una estrategia posible es tomar 10 medidas y quedarse con las 3 mejores (menor coste en ciclos por iteración) y hacer la media geométrica de estos tres valores.

Calentamiento: para evitar efectos de inicialización de las cachés, de las TLBs, etc, es interesante realizar una

fase de calentamiento antes de tomar las medidas. Para ello inicializar los datos del vector A[] que se van a leer en el bucle de computación. Para la inicialización de los datos es interesante utilizar valores aleatorios, pero intentando que no se produzcan desbordamientos en la representación *Double*. Una forma de evitarlo es que se usen valores iniciales con valor absoluto acotado en el intervalo [1,2) con signo positivo o negativo de forma aleatoria.

Procedimiento de ejecución: en general es mejor ejecutar un experimento individual cada vez. Así reducimos la probabilidad de resultados distorsionados por el efecto del planificador del sistema operativo.

Comentarios generales: Para hacer estos experimentos es imprescindible conocer la jerarquía de memoria caché del procesador en detalle. Hay diferentes posibles formas de averiguar esta información.

Plataforma de ejecución: Realizar las ejecuciones de todos los experimentos en el mismo ordenador que puede ser vuestro portátil o el ordenador del aula. Utilizar el sistema operativo Linux y el compilador gcc con la opción -O0 (no optimización).

LO QUE HAY QUE ENTREGAR: debe entregarse un único informe de la práctica por cada pareja de prácticas **a través del campus virtual**. El informe debe seguir el esquema para el que se proporcionará una plantilla en Word y en Latex. En la plantilla se pueden cambiar los nombres de las secciones pero debe haber necesariamente un título, autores, resumen, introducción, ... tal como se aprendió en la asignatura Fundamentos de Computadores. Es fundamental describir el procesador que se utiliza para los experimentos y su jerarquía de memoria. El informe debe contener también el listado completo del código comentado (no es necesario incluir el código interno de las rutinas de medida de ciclos). También se deben incluir explicaciones sobre los experimentos realizados, las gráficas obtenidas y una interpretación de los resultados. Como en todo informe, al final deben presentarse las conclusiones.

CRITERIOS DE EVALUACIÓN: cumplimiento de las especificaciones del enunciado de la práctica, calidad de los resultados presentados y de la memoria. Esto incluye rigurosidad, claridad del informe y calidad de la presentación de resultados. También se tendrá en cuenta la autonomía en el desarrollo del trabajo, la actitud frente al trabajo y la calidad de las respuestas si el profesor pide aclaraciones. La nota será individual para cada miembro del grupo. Obviamente la copia de programas o informes de cualquier fuente se considerará motivo de suspenso. Atención: entregar el informe fuera de plazo implicará una penalización en la evaluación de la práctica.