Arquitectura de Computadores

2º Curso Grao Enx. Informática

Práctica 1

Xerarquía de Memoria Caché: Estudo do Efecto da Localidade das Referencias a Memoria nas Prestacións dos Programas en Microprocesadores

Obxectivos: mediante a realización de programas de proba en linguaxe C, caracterizar o coste temporal dun bucle de computación, variando diferentes parámetros no patrón de acceso e no tamaño do conxunto de datos, e observar e interpretar o efecto do sistema de memorias caché do microprocesador tendo en conta os conceptos de localidade espacial e temporal e precarga (prefetching).

Equipos: grupos de prácticas de ata dúas persoas.

Sistema para a toma de datos: ordenadores da aula de informática.

Tempo de traballo presencial: 4 sesións.

Prazo de entrega: Venres, 18 de Marzo 2016 (incluido).

Forma de entrega e formato: en papel, cunha grapa na parte superior esquerda. Entregalo persoalmente ao profesor (en clase ou no despacho 111 do CITIUS) ou deixalo no casilleiro de correo do profesor na portería do CITIUS.

Valoración: 40% da nota de prácticas (até 1.6 puntos sobre 10 na nota total).

Apartado 1

Facer un programa en C que reserve memoria dinámica para un vector (*array*) A[] de números representados en punto flotante tipo *Double*. Levar a cabo unha reducción de suma de punto flotante sobre os R elementos do vector A[0], A[D], A[2D], A[3D],....,A[(R-1)*D] (é dicir sumar este R elementos almacenando o valor nunha variable tipo *Double*). D é un parámetro que vamos a variar para estudar o efecto da localidade. Para que os resultados sexan consistentes co apartado 2, a referencia aos elementos de A[] debe facerse de xeito indirecto a través dun vector de enteiros e[], é dicir facer referencias do tipo A[e[i]], con e[i]=0, D, 2D, 3D, etc. Por consistencia co apartado 2 o vector e[] debe ter 10 R índices, con 10 secuencias de R índices do tipo 0, D, 2D, 3D, etc.

Repetir 10 veces a computación da reducción. Logo de cada unha das computacións, almacenar o resultado nun elemento distinto dun vector S[] de 10 elementos tipo *Double*. Para evitar optimizacións non desexadas

do compilador, é conveniente imprimir ao final os elementos do vector S[] (resultados).

Tomar a medida de ciclos (utilizando as rutinas que aparecen no programa C adxunto) totais das 10 repeticións (non incluír a impresión dos resultados nesta medida), e obter o número de ciclos medio por cada iteración da computación total.

Repetir o experimento para diferentes valores de R e D.

Valores de R: para un valor específico de D, é necesario realizar un total de 7 medidas. En cada medida seleccionamos o valor de R de tal xeito que as **lecturas do vector A[] correspondan a L liñas caché diferentes**. Tomar medidas de ciclos para os seguintes valores de L: 0.5*S1, 1.5*S1, 0.5*S2, 0.75*S2, 2*S2, 4*S2, 8*S2 , donde S1 é o número de liñas cache que collen na cache de datos de nivel 1, e S2 é o número de liñas cache que collen na cache de nivel 2. Será necesario investigar cales son os valores de S1 e S2. É necesario estar seguro de que estes valores son os correctos xa que determinarán a calidade dos experimentos.

Valores de D: para cada valor de L, tomaremos cinco medidas, variando o valor de D entre 1 e 100. Os valores de D deben seleccionarse de tal xeito que os experimentos realizados permitan estudar de forma representativa os diferentes efectos da localidade de acceso aos datos. Por exemplo, no tería sentido utilizar D=90, 91, 92, 93, 94 e 95, xa que é probable que os resultados sexan moi semellantes, e polo tanto non permitirían interpretar outros efectos da localidade que poidan ser de interese. **A selección dos valores de D axeitados é un parte moi importante do experimento**.

Número de elementos do vector A[]: determinado polos valores seleccionados de R e D. Para a reserva de memoria resulta conveniente alinear o comenzo do vector co comenzo dunha liña cache. Unha alternativa para facer isto é utilizar a función **_mm_malloc(TC,CLS)**, donde TC é o número de bytes a reservar e CLS é o tamaño de liña cache en bytes. Esta función devolve un punteiro aliñado a CLS bytes. Para liberar a memoria utilizar a correspondente función **_mm_free(P)**, donde P é o punteiro que apunta ao comenzo do vector de memoria reservado. Para utilizar estas funcións é necesario poñer esta liña no arquivo: **#include <pmmintrin.h>**. Na compilación con gcc é necesario utilizar a opción **-msse3**.

Presentación de resultados: facer unha representación gráfica do coste medio por iteración da computación (eixo y) vs. o número total de liñas cache diferentes que se referencian (eixo x) para os diferentes valores de D. Incorporar tamén táboas con datos se é necesario.

Interpretación dos experimentos: facer unha interpretación dos resultados en relación aos conceptos de localidade temporal e espacial en memorias cache. Busca información sobre os mecanismos de precarga (prefetching) en caches, xa que pode ser de utilidade para interpretar os resultados.

Variabilidade dos resultados: é recomendable tomar varias medidas para un mesmo experimento para observar a variabilidade. Unha estratexia posible é tomar 10 medidas e quedarse coas 3 mellores (menores coste en ciclos por iteración) e facer a media xeométrica destes tres valores.

Quencemento: para evitar efectos de inicialización das cachés, das TLBs, etc, é interesante realizar un fase de quencemento antes de tomar as medidas. Para o quencemento simplemente inicializar os datos do vector A[]

que se van a ler no bucle de computación. Para a inicialización dos datos é interesante utilizar valores aleatorios, pero intentando que non se produzan desbordamentos na representación *Double*. Unha posibilidade sinxela é inicializar os datos de tal xeito que o seu valor absoluto esté acotado no intervalo [1,2) e con signo positivo ou negativo de xeito aleatorio.

Procedemento de execución: en xeral é mellor executar un experimento individual cada vez. Así reducimos a probabilidade de resultados distorsionados polo efecto do planificador do sistema operativo. É mellor minimizar os programas activos durante a execución dos experimentos.

Plataforma de execución: os datos dos experimentos deben obterse cos ordenadores da aula de informática na que levamos a cabo as prácticas, co sistema operativos Linux e o compilador GCC coa opción -O0 (non optimización). Non obstante, é posible desenvolver e depurar o programa en calquera outra plataforma.

Apartado 2

Repetir o experimento anterior, pero forzando un patrón de acceso aleatorio. Debe accederse aos memos datos que no experimento anterior, pero nunha orde aleatoria. Utilizar un vector de indices enteiros e[] para o acceso indirecto do elementos do vector A[]. En cada unha das 10 veces que se repite a computación o patrón de acceso será distinto (con alta probabilidade), polo que o vector e[] debe ter 10 R índices, correspondente a 10 secuencias aleatorias de R índices.

MEMORIA DA PRÁCTICA: debe entregarse unha memoria (en papel) cos resultados da práctica. A memoria debe conter unha introdución, que describa o obxectivo dos experimentos e a xerarquía de memoria do procesador (co maior número de datos posible do procesador e o sistema). Para cada apartado debe conter o listado completo do código (non é necesario incluir o código interno das rutinas de medida de ciclos), coas explicacións necesarias, as gráficas obtidas, e unha interpretación das mesmas. Finalmente deben presentarse as conclusións finais a modo de resumo.

CRITERIOS DE AVALIACIÓN: cumprimento das especificacións do enunciado da práctica, rigurosidade no plantexamento e interpretación de resultados, estudo autónomo e presentación de resultados (calidade da memoria). Tamén se terá en conta o traballo realizado na aula durante as sesións de prácticas e a calidade das respostas ante as aclaracións requeridas. A nota será individual para cada membro do grupo.