# Arquitectura de Computadores

2º Curso Grao Enx. Informática

# Práctica 2

# Programación Multinúcleo y extensiones SIMD

**Objetivos:** programar un algoritmo simple con matrices en punto flotante, utilizando diferentes grados de optimización (utilización de varios núcleos, utilización de extensiones vectoriales SIMD, estrategias para reducir los fallos cache,..) y tomar medidas de rendimiento combinando diferentes optimizaciones y tamaños de problema.

**Equipos:** grupos de prácticas de hasta dos personas.

**Plazo de entrega:** Viernes 7 de Mayo 2020 hasta las 2 de la tarde.

**Valoración:** la máxima valoración que se podrá conseguir en cada apartado (sobre 10) es la siguiente: apartados i) y ii) → 3 puntos, apartado iii) → 3.5 puntos, apartado iv) → 3.5 puntos.

**Descripción:** Hacer diferentes programas en C que realicen la computación del vector de salida **e** utilizando el siguiente pseudocódigo de partida:

**Entradas:**

**a**[N][8],**b**[8][N],**c**[8]: matrices y vector que almacenan valores aleatorios de tipo double.

**Salida:**

**f**: variable de salida tipo double

**Computación:**

```
d[N][N]=0; // inicialización de todas las componentes de d a cero;
 for (i=0; i<N; i++) {
    for(j=0; j<N; j++) {
       for (k=0; k<8; k++) {
         d[i][j] += 2 * a[i][k] * ( b[k][j]- c[k]);
       }
     }
ind[N]: vector desordenado aletoriamente que contiene índices de fila/columna
f=0;
for (i=0;i<N;i++){
    e[i]= d[ind[i]][ind[i]]/2;
    f+=e[i];}
Imprimir el valor de f
```

Las diferentes versiones son las siguientes:

i) Programa secuencial base (codificación en C del pseudocódigo anterior)

ii) Programa secuencial optimizado: intentar modificar el código de modo que se obtenga al final el mismo vector resultado pero que se reduzca el tiempo de ejecución. Para ello probar a realizar las mejoras que cada uno considere oportunas. Ejemplo de posibles optimizaciones: cambiar el código de manera que se realicen las operaciones en otro orden, cambiar el orden de los lazos del producto de matrices para acceder con más localidad a ellas, realizar *unrolling* (desenrrollamiento de lazos , realizar operaciones por bloques, etc)

iii) Programa secuencial optimizado utilizando procesamiento vectorial SIMD. Para ello utilizar extensiones SSE3.

iv) Programa utilizando OpenMP para paralelizar la versión secuencial optimizada (no está permitido utilizar variables tipo "reduction"), sin utilizar extensiones vectoriales, y variando el número de hilos hasta el máximo número de cores del PC en que estés haciendo los experimentos: 1, 2, 4, 6 y 8.

Los códigos de cada apartado deben compilarse del siguiente modo:

i) El código secuencial deberá compilarse sin optimizaciones del compilador: "`gcc -O0 p2_apartado1.c`", y con optimización incluída la autovectorización "`gcc -O2 p2_apartado1.c`"

ii) El código secuencial mejorado deberá compilarse sin optimizaciones del compilador: "`gcc -O0 p2_apartado2.c`"

iii) Estos códigos deberán compilarse sin optimizaciones del compilador: "`gcc -O0 p2_apartado3.c`". Añadir en la cabecera del código "`#include <immintrin.h>`" para que el compilador reconozca las instrucciones vectoriales SSE3.

iv) Estos códigos deberán compilarse sin optimizaciones del compilador y con el flag de OpenMP: "`gcc -O0 -fopenmp p2_apartado4.c`" y también con la autovectorización: "`gcc -O2 -fopenmp p2_apartado4.c`".

Hacer experimentos considerando que el número N de filas y columnas de la matriz toma los valores siguientes: **N=250, 500, 750, 1000, 1500, 2000, 2550, 3000.** En todos los casos hacer reserva de memoria dinámica de las matrices y vectores e inicializarlos con valores aleatorios en un intervalo prefijado. **Medir el número de ciclos** de la parte del programa en que se hace la computación indicada en el pseudocódigo. **Asegurarse de que el resultado final en todas las**

**versiones es el mismo, es decir, que todas las versiones realizan la computación correctamente.** Cuando se utilicen OpenMP o extensiones SSE3 se debe incluir en la medida de tiempo todo lo que implique una sobrecarga respecto del programa secuencial optimizado. Para cada caso, tomar al menos 10 medidas, y seleccionar la mediana de estos valores como valor final de medida de tiempo de ejecución.

**Cada uno tomará las medidas en su propio ordenador. Se deben indicar en la memoria el tipo de procesador y sus características principales (jerarquía de memoria, procesador, número de cores, tamaño de memoria RAM,...).**

**MEMORIA DE LA PRÁCTICA Y CÓDIGOS FUENTE:** debe entregarse una memoria de la práctica por cada grupo de dos personas y la entregará cualquiera de los miembros del grupo. Será un documento en PDF siguiendo la misma plantilla que se usó para la práctica 1 (título, autores, resumen, introducción, descripción de los experimentos, resultados y su análisis, conclusiones y bibliografía). Hay que explicar lo más relevante de los códigos asociados a cada apartado, incluir gráficas de resultados e interpretación de dichos resultados. La memoria debe incluir las gráficas que se consideren oportunas para sacar conclusiones de los resultados obtenidos. **Es de especial interés representar la ganancia en velocidad (también llamada aceleración o *speedup*) de la versión secuencial optimizada con respecto a la versión inicial compilada con -O0, la ganancia en velocidad de los códigos de los apartados iii) y iv) con respecto a la versión secuencial optimizada, y finalmente la versión inicial compilada con -O3 respecto de todas las demás). En el caso del apartado iv) además de las gráficas que consideres adecuado es imprescindible representar en una única gráfica la ganancia en velocidad conseguida para los diferentes números de hilos variando el valor de N y una gráfica separada mostrando el comportamiento para el valor más grande de N.**

**Además de la memoria en pdf se subirá a la tarea del campus virtual correspondiente un archivo zip o rar que incluirá los códigos fuente completos de los diferentes apartados. El envío del código es obligatorio** (y dentro del plazo de entrega de la memoria) y solo se subirá un ejemplar por cada pareja de prácticas. Los códigos deben estar todos dentro de un directorio que se llame *p2_ac_21*. Es importante seguir el criterio de nombres de archivo especificado en el guión.

**CRITERIOS DE EVALUACIÓN:** cumplir las especificaciones del enunciado de la práctica buscando siempre obtener la mínima latencia (tiempo de ejecución) del programa en las diferentes

implementaciones, rigurosidad en el planteamiento e interpretación de resultados, exploración de alternativas, estudo autónomo e presentación de resultados (calidad de la memoria).