

Parte II

PROGRAMACIÓN PARALELA UTILIZANDO EL JUEGO DE INSTRUCCIONES DEL PROCESADOR

3. Optimización basada en el juego de instrucciones vectoriales Intel SSE2

3.1. Objetivos

El uso eficiente de todo el juego de instrucciones de un procesador es un factor determinante en el rendimiento de un programa. En particular, el desarrollo de programas que utilizan las extensiones SIMD/vectoriales de los procesadores modernos permite explotar la organización SIMD (Single Instruction Multiple Data) existente en algunas unidades funcionales del procesador. El objetivo de esta práctica es describir el juego de instrucciones Intel SSE2 (Intel Streaming SIMD Extensions 2) disponible en los procesadores Intel y AMD actuales, e introducir al estudiante en su manejo mediante la librería *intrinsics* que permite el uso de instrucciones SIMD/vectoriales dentro de programas C. A través de la optimización de diversos casos de estudio, se pretende no sólo que el estudiante se familiarice con los problemas habituales de este modelo de procesamiento sino también que desarrolle la capacidad de hacer un análisis crítico de los resultados obtenidos al optimizar los diferentes códigos.

3.2. Metodología

3.2.1. Formato de almacenamiento comprimido (matrices dispersas)

Utilizando el código de la Figura 3 para el cálculo de una norma para cada fila de una matriz almacenada en formato comprimido que fue utilizado en la práctica 2 de la asignatura, desarrollar los siguientes programas que explotan el juego de instrucciones SSE2 mediante llamadas a funciones de la librería *intrinsics*:

- Optimizar la operación de multiplicación del bucle interno utilizando el tipo de datos *double* (números en punto flotante de doble precisión).

```

for (i=0; i<nfilas; i++) {
    normafila = 0;
    for (j=0; j<dimfilas[i]; j++) {
        normafila += valores[i][j] * valores[i][j];
    }
    norma[i] = sqrt(normafila);
}

```

Figura 3: Cálculo de la norma de cada fila de una matriz en formato comprimido.

- Optimizar la operación de multiplicación del bucle interno utilizando el tipo de datos *float* (números en punto flotante de simple precisión).

Medir el aumento de rendimiento de las dos versiones optimizadas con SSE2 respecto a la versión secuencial. Rellenar las siguientes tablas con los valores proporcionados por los programas:

| nfilas x ncols | Tiempo secuencial | Tiempo SSE2 (double) | Aceleración |
|----------------|-------------------|----------------------|-------------|
| 15000 x 15000 | | | |
| 15000 x 20000 | | | |
| 15000 x 25000 | | | |

| nfilas x ncols | Tiempo secuencial | Tiempo SSE2 (float) | Aceleración |
|----------------|-------------------|---------------------|-------------|
| 15000 x 15000 | | | |
| 15000 x 20000 | | | |
| 15000 x 25000 | | | |

Explicar los resultados obtenidos desde el punto de vista del uso eficiente de la jerarquía de memoria del sistema computador (registros vectoriales, memoria cache, memoria RAM y almacenamiento secundario). Utilizar el comando de unix *top* para analizar el intercambio de información entre la memoria RAM y el almacenamiento secundario (*swap*) durante la ejecución de los programas.

3.2.2. Formato de almacenamiento no comprimido (matrices densas)

Utilizando el código de la Figura 4 para el cálculo de una norma para cada fila de una matriz almacenada en formato no comprimido (matriz densa), desarrollar los siguientes programas que explotan el juego de instrucciones SSE2 mediante llamadas a funciones de la librería *intrinsics*:

- Optimizar la operación de multiplicación del bucle interno utilizando el tipo de datos *float* (números en punto flotante de simple precisión). Nótese que el código optimizado es casi idéntico al desarrollado con matrices en formato comprimido.
- Optimizar las tres operaciones aritméticas (multiplicación, suma y raíz cuadrada) que aparecen durante la ejecución del bucle más externo. Nótese que

```

for (i=0; i<nfilas; i++) {
    normafila = 0;
    for (j=0; j<ncols; j++) {
        normafila += valores[i][j] * valores[i][j];
    }
    norma[i] = sqrt(normafila);
}

```

Figura 4: Cálculo de la norma de cada fila de una matriz en formato no comprimido.

es importante recordar las técnicas descritas durante el seminario sobre como realizar 4 productos de vectores de 4 elementos utilizando instrucciones SIMD/vectoriales.

Como el caso anterior, medir el aumento de rendimiento de las dos versiones optimizadas con SSE2 respecto a la versión secuencial y rellenar la siguiente tabla con los valores proporcionados por los programas.

| nfilas x ncols | Tiempo secuencial | Versión 1 con SSE2 | | Versión 2 con SSE2 | |
|----------------|----------------------|--------------------|-------------|--------------------|-------------|
| | | Tiempo | Aceleración | Tiempo | Aceleración |
| 15000 x 15000 | | | | | |
| 15000 x 20000 | | | | | |
| 15000 x 25000 | | | | | |

Finalmente, explicar la influencia que el uso de la jerarquía de memoria tiene sobre los distintos rendimientos medidos.

3.3. Material complementario

- Código secuencial `practica3a.c` para el cálculo de la norma de cada fila de una matriz almacenada en formato comprimido en la que los elementos de la matriz son de tipo *double*.
- Código secuencial `practica3b.c` para el cálculo de la norma de cada fila de una matriz almacenada en formato comprimido en la que los elementos de la matriz son de tipo *float*.
- Código secuencial `practica3c.c` para el cálculo de la norma de cada fila de una matriz almacenada en formato denso en la que los elementos de la matriz son de tipo *float*.
- Fichero *Makefile* para la compilación de los tres códigos descritos anteriormente.
- Referencia completa de la librería *intrinsics*. Consultar URL http://download.intel.com/support/performance/c/linux/v9/intref_cls.pdf.

3.4. Evaluación

La evaluación se realizará por el procedimiento de defensa oral, con especial énfasis en los detalles de manejo de las instrucciones SSE2 por parte del programador. El estudiante podrá obtener una bonificación sobre la nota final de la asignatura si entrega la práctica antes de las fechas indicadas en la siguiente tabla:

| Grupo prácticas | Fecha de entrega |
|-----------------|------------------|
| Jueves | 5 marzo 2009 |
| Viernes | 6 marzo 2009 |