

# 66:20 Organización de Computadoras

## Trabajo práctico 2: Multiplicación de Matrices

### 1. Objetivos

Familiarizarse con las técnicas de optimización de uso de la memoria cache, escribiendo un programa portable que resuelva el problema descrito en la sección 6.

### 2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

### 3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes.

Además, es necesario que el trabajo práctico incluya (entre otras cosas, ver sección 8), la presentación de los resultados obtenidos, explicando, cuando corresponda, con fundamentos reales, las causas o razones de cada resultado obtenido.

El informe deberá respetar el modelo de referencia que se encuentra en el grupo, y se valorarán aquellos escritos usando la herramienta  $\text{\TeX}$  /  $\text{\LaTeX}$ .

### 4. Introducción

La manera obvia de implementar la multiplicación de matrices en lenguaje C es sencilla, pero terriblemente ineficiente en cuanto a la utilización de la memoria cache:

```
for(i=0;i<N;i++)
  for(j=0;j<M;j++)
    for(k=0;k<L;k++)
      c[i][j] += a[i][k] * b[k][j];
```

Este programa multiplica las matrices  $A \in \mathbb{R}^{N \times L}$  y  $B \in \mathbb{R}^{L \times M}$ , dando como resultado una matriz  $C \in \mathbb{R}^{N \times M}$ , pero al recorrer una de las matrices por filas y la otra por columnas, para matrices mayores que la memoria cache vamos a tener conflictos.

## 5. Recursos

En la sección de archivos de la materia [1] está disponible el esqueleto del trabajo, incluyendo código que valida la multiplicación de matrices, además de evaluar el tiempo que tarda para determinados tamaños de matriz. Una herramienta que puede ser útil es `cachegrind`, que forma parte de las herramientas Valgrind [2].

## 6. Programa

Se trata de optimizar la función `matmult_cuadrado()`, que guarda en la matriz  $C \in \mathbb{R}^{M \times M}$  el resultado de multiplicar las matrices A y B, también de  $M \times M$  elementos.

## 7. Implementación

El programa a implementar deberá satisfacer algunos requerimientos mínimos, que detallamos a continuación.

- El prototipo de la función deberá ser el siguiente:

```
void matmult_cuadrado(const int M, const double *A, const double *B, double *C)
```

- La función deberá satisfacer la validación provista
- La función deberá intentar implementar algún tipo de mejora de la performance

### 7.1. Algoritmo

Se provee una primera optimización, que consiste en dividir la matriz en bloques que sean más manejables. Ver [3].

## 8. Informe

El informe deberá incluir:

- Este enunciado;

- Documentación relevante al diseño e implementación del programa, en particular las optimizaciones que se intentaron;
- Los resultados obtenidos y un análisis de los mismos;
- El código fuente completo, en dos formatos: digitalizado<sup>1</sup> e impreso en papel.

## 9. Fecha de entrega

La última fecha de entrega y presentación es el jueves 6 de Diciembre de 2007.

## Referencias

- [1] Código fuente con el esqueleto del trabajo práctico, <http://groups.yahoo.com/group/orga6620/files/Trabajos%20Pr%80%A0%A0%E1cticos/matmult.tgz>.
- [2] Herramienta de profiling Valgrind, <http://valgrind.org/>
- [3] The Cache Performance and Optimizations of Blocked Algorithms. Monica S. Lam and Edward E. Rothberg and Michael E. Wolf. Computer Systems Laboratory, Stanford University. <http://suif.stanford.edu/papers/lam91.ps>

---

<sup>1</sup>No usar diskettes: son propensos a fallar, y no todas las máquinas que vamos a usar en la corrección tienen lectora. En todo caso, consultá con tu ayudante.