



Presentación

Objetivos

Esta práctica tiene como objetivo el diseño e implementación de una aplicación concurrente multi-hilo sincronizada utilizando pthreads. Para su realización se pondrán en práctica muchos de los conceptos presentados en esta asignatura.

Presentación de la práctica

Hay que presentar los archivos con el código fuente realizado. Toda la codificación se hará exclusivamente en lenguaje C.



Enunciado de la práctica

El objetivo de esta práctica es implementar la multiplicación de matrices de forma concurrente. Vamos a implementar dos métodos de implementar la multiplicación:

- El método estándar, multiplicando las filas por las columnas
- El método Strassen, que se basa en un paradigma divide-y-vencerás recursivo.

Multiplicación matrices tradicional

Este algoritmo está compuesto por tres bucles. El primer bucle recorre las filas de la matriz de resultado, en el segundo bucle se recorre las columnas de la matriz resultado y en el tercer bucle se realiza la multiplicación de toda una fila de la primera matriz por una columna de la segunda matriz, guardando en resultado en la celda correspondiente de la matriz resultado. La complejidad de tiempo de este algoritmo es $O(N^3)$:

$$\begin{array}{c} \text{row } i \\ \left[\begin{array}{c} \boxed{} \\ \\ \end{array} \right] \times \left[\begin{array}{c} \boxed{} \\ \\ \end{array} \right] = \left[\begin{array}{c} \boxed{} \\ \\ \end{array} \right] \\ \begin{array}{ccc} A & \times & B \\ m \times n & & n \times p \end{array} = \begin{array}{c} C \\ m \times p \end{array} \end{array}$$

$c_{ij} = \text{row } i \times \text{col } j = \sum_{k=1}^n a_{ik} b_{kj}$

Multiplicación matrices Divide-y-Vencerás

El método simple basado en el paradigma Divide-y-vencerás para multiplicar dos matrices cuadradas (cuyas dimensiones son múltiplo de 2^k) consiste en realizar los siguientes pasos¹:

1. Dividir las matrices A y B en 4 submatrices de tamaño $N/2 \times N/2$ como se muestra en el siguiente diagrama.
2. Calcular los siguientes valores de forma recursiva. $ae + bg$, $af + bh$, $ce + dg$ y $cf + dh$.

¹ <https://www.geeksforgeeks.org/strassens-matrix-multiplication/>



SISTEMAS CONCURRENTES Y PARALELOS

PRÀCTICA 4

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

A B C

A, B and C are square matrices of size $N \times N$
a, b, c and d are submatrices of A, of size $N/2 \times N/2$
e, f, g and h are submatrices of B, of size $N/2 \times N/2$

Con este método, se realizan 8 multiplicaciones para matrices de tamaño $N/2 \times N/2$ y 4 sumas. La suma de dos matrices lleva $O(N^2)$ tiempo. Entonces la complejidad del tiempo se puede escribir como: $T(N) = 8T(N/2) + O(N^2) = O(N^3)$.

Multiplicación matrices Divide-y-Vencerás: método Strassen

La idea del **método de Strassen** es reducir el número de llamadas recursivas a 7. El método de Strassen es similar al método simple de dividir y conquistar anterior en el sentido de que este método también divide matrices en submatrices de tamaño $N/2 \times N/2$ como se muestra en el diagrama anterior, pero en el método de Strassen, las cuatro submatrices de resultado se calculan utilizando las siguientes fórmulas:

$$\begin{aligned} p1 &= a(f - h) & p2 &= (a + b)h \\ p3 &= (c + d)e & p4 &= d(g - e) \\ p5 &= (a + d)(e + h) & p6 &= (b - d)(g + h) \\ p7 &= (a - c)(e + f) \end{aligned}$$

The $A \times B$ can be calculated using above seven multiplications.
Following are values of four sub-matrices of result C

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} p5 + p4 - p2 + p6 & p1 + p2 \\ p3 + p4 & p1 + p5 - p3 - p7 \end{bmatrix}$$

A B C

A, B and C are square matrices of size $N \times N$
a, b, c and d are submatrices of A, of size $N/2 \times N/2$
e, f, g and h are submatrices of B, of size $N/2 \times N/2$
p1, p2, p3, p4, p5, p6 and p7 are submatrices of size $N/2 \times N/2$

El método de Strassen tiene una complejidad de: $T(N) = 7T(N/2) + O(N^2) = O(N^{2.807})$.

El inconveniente de este método es que, debido a la precisión limitada de la aritmética informática en valores no enteros, a medida que las matrices son más grandes se generan pequeños errores de precisión que al acumularse debido a la recursividad provoquen el resultado no sea el mismo que el obtenido con el método tradicional.

El objetivo de esta práctica es que implementes la versión concurrente de la multiplicación de matrices, tanto para el método tradicional como para el método de Strassen. En la implementación concurrente se tiene que garantizar que la aplicación es determinista y que no contiene condiciones de carrera. Para ello hay que utilizar los mecanismos de sincronización de pthreads vistos en clase.



SISTEMAS CONCURRENTES Y PARALELOS

PRÀCTICA 4

La aplicación concurrente recibirá como parámetros los ficheros de entrada con las dos matrices a multiplicar, el fichero de salida donde se tiene que guardar la matriz resultado y el número de hilos que hay que utilizar:

Sintaxis: MultMat_Conc <MatrixA_File> <MatrixB_File> <ResultMatrix_File> <#Threads>

Como resultado el programa generará dos ficheros con la matriz resultado, uno por cada una de las versiones. El fichero con la extensión std contendrá la matriz resultado para el método tradicional y el fichero con la extensión str contendrá la matriz resultado para el método strassen.

• Formato ficheros de las matrices.

Los ficheros de matrices utilizados en el programa tienen el formato que se muestran en los siguientes ejemplos. Un entero que indica la dimensión de la matriz y a continuación los valores de la matriz por filas.

Ejemplo:

Matriz A 4x4	Matriz B 4x4	Matriz Resultado 4x4
4 -5.00 -3.68 2.56 -0.41 0.33 -2.81 -4.53 1.79 1.79 4.35 -1.16 0.19 3.31 -4.65 -4.47 0.30	4 1.71 -4.92 -1.17 -4.33 -0.83 1.87 0.89 4.30 3.46 0.27 -4.08 1.54 -0.84 2.01 4.10 2.62	4 3.71 17.59 -9.55 8.69 -14.28 -4.50 22.93 -15.80 -4.72 -0.60 7.29 9.67 -6.20 -25.58 11.46 -40.43



Funciones involucradas.

En la práctica podéis utilizar las siguientes funciones de c, entre otras:

- `int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine) (void *), void *arg);`
- `int pthread_join(pthread_t thread, void **retval);`
- `void pthread_exit(void *retval);`
- `int pthread_cancel(pthread_t thread);`



Análisis prestaciones

Calcular el tiempo de ejecución de la versión concurrente y compararlo con el de la versión secuencial. Discutir los resultados obtenidos. El objetivo de la práctica es que **la versión concurrente sea más rápida que la versión secuencial**, cuando se calculen matrices grandes (2048x2048).

Analizar también, el tiempo de ejecución de la versión concurrente en función del número de hilos. Entregar un pequeño informe en donde se muestren (preferiblemente de forma gráfica) y expliquen los resultados obtenidos.

Indicar en el informe las características del hardware en donde habéis obtenido los resultados (especialmente el número de procesadores/núcleos).



Evaluación

La evaluación de la práctica se realizará en función del grado de eficiencia y concurrencia logrado.

Se utilizarán los siguientes criterios a la hora de evaluar las dos implementaciones de la práctica, partiendo de una nota base de 10:

- Aspectos que se evalúan para cada una de las versiones:
 - Ejecución Secuencial (Susp)
 - Tener condiciones de carrera ([-2.0p,...,-5.0p])
 - Prestaciones Aplicación concurrente (-2.0p, +0.5p)
 - Métodos no implementados concurrentemente: tradicional o Strassen (-3.0p cada uno)
 - No realizar Join hilos (-1.0p)
 - Control Errores: No finalizar de forma controlada los hilos (-0.75p)
 - Descomposición no balanceada (Versión óptima) (-1.0p)
 - Descomposición parcial: No se asigna todo el trabajo a los hilos (-1.0p)
- Evaluación del informe de la práctica:
 - + Informe completo, con gráficos y conclusiones correctas (+0.25p,+1.0p)
 - Sin informe (-1.0p)
 - Informe con incoherencias en los tiempos sin justificar (-0.5p)
 - Informe: análisis poco riguroso (-0.75p)
 - No se explica el diseño de la solución propuesta (-0.75p)
 - Descripción de los problemas encontrados y como se han solventado.
- Estilo del código.
 - Comentarios
 - Utilizar una correcta indentación
 - Descomposición Funcional (Código modular y estructurado)
 - Control de errores.
 - Test unitarios para verificar la correcta ejecución de la lógica de la aplicación.

Se puede tener en cuenta criterios adicionales en función de la implementación entregada.



Versiones Secuenciales



SISTEMAS CONCURRENTES Y PARALELOS

PRÀCTICA 4

Junto con el enunciado se os proporciona una versión secuencial en C para resolver el problema descrito. El código secuencial está basado en CLion y que utiliza cmake para la compilación.



Formato de entrega

MUY IMPORTANTE: La entrega de código que no compile correctamente, implicará suspender TODA la práctica.

No se aceptarán prácticas entregadas fuera de plazo (salvo por razones muy justificadas).

La entrega presencial de esta práctica es obligatoria para todos los miembros del grupo.

Comenzar vuestros programas con los comentarios:

```
/* -----  
Práctica 4.  
Código fuente: MultMat_Conc.c  
Grau Informàtica  
NIF i Nombre completo autor1.  
NIF i Nombre completo autor2.  
----- */
```

Para presentar la práctica dirigiros al apartado de Actividades del Campus Virtual de la asignatura de Sistemas Concurrentes y Paralelos, ir a la actividad "Práctica 4" y seguid las instrucciones allí indicadas.

Se creará un fichero tar/zip con todos los ficheros fuente de la práctica, con el siguiente comando:

```
$ tar -zcvf prac1.tgz fichero1 fichero2 ...
```

se creará el fichero "prac1.tgz" en donde se habrán empaquetado y comprimido los ficheros "fichero1", fichero2, y ...

Para extraer la información de un fichero tar se puede utilizar el comando:

```
$ tar -zxvf tot.tgz
```

El nombre del fichero tar tendrá el siguiente formato: "Apellido1Apellido2PRA4.tgz". Los apellidos se escribirán sin acentos. Si hay dos autores, indicar los dos apellidos de cada uno de los autores separados por "_". Por ejemplo, el estudiante "Perico Pirulo Palotes" utilizará el nombre de fichero: PiruloPalotesPRA4.tgz

Importante: Antes de subir vuestro código eliminar todas las matrices de las carpetas Input, Results y Test. Si el archivo subido supera los 20 MBs no se aceptará.



Fecha de entrega

Entrega a través de Sakai el 18 de diciembre, entrega presencial en la clase de grupo de laboratorio durante la semana del 18 al 22 de diciembre de 2023.

