Multiplicación de Matrices con OpenMP

STUDENTS:

Johan Sebastian Méndez Ibarra

PROFESSOR:

Jhon Corredor Franco

COURSE:

Introducción a los Sistemas Distribuidos

Bogotá D.C.
2025

Índice

1. Resumen

Se revisó, corrigió y documentó el proyecto de Multiplicación de Matrices (MM) en C con paralelismo `OpenMP`.

Archivos modificados (únicamente estos):

- `mmClasicaOpenMP.c`: correcciones funcionales y medición de tiempo.
- `Makefile`: compilación correcta con `-fopenmp` y reglas mínimas.
- `auto.txt`: script en Perl para automatizar el plan de pruebas.

2. Requisitos del problema

De acuerdo con lo establecido por el profesor, el trabajo requiere la corrección y documentación del código en C para la multiplicación de matrices con `OpenMP`, la adaptación del `Makefile` para garantizar una compilación adecuada con soporte de paralelismo, y la modificación del script de automatización de pruebas. Asimismo, debe diseñarse un plan de experimentación con doce tamaños de matriz menores a 14 000 y configuraciones de paralelismo de {1, 4, 8, 16, 20} hilos, incorporando un número suficiente de repeticiones que permitan obtener resultados consistentes y con sustento estadístico frente al ruido del sistema operativo.

3. Cambios por archivo

3.1. `mmClasicaOpenMP.c`

Qué se corrigió/añadió:

- Paralelización con `#pragma omp parallel for` (bucle por filas).
- Orden de bucles `i-k-j` para mejorar localidad en B.
- Medición de tiempo con `omp_get_wtime()` a través de `InicioMuestra()` y `FinMuestra()`.
- Argumentos de ejecución: `N [hilos]`. Si se pasa `hilos`, se fija `omp_set_num_threads(hilos)`.
- Salida en CSV de una línea: `N,threads,time_sec`, útil para análisis.

Fragmento relevante (interface & salida):

```
int main(int argc, char **argv){
    // Uso: ./mmClasicaOpenMP N [hilos]
    // Imprime: N,threads,time_sec
    ...
    printf("%d,%d,%.6f\n", N, threads, dt);
    ...
}
```

Listing 1: Interfaz de ejecución y salida CSV

3.2. `Makefile`

Qué se corrigió/añadió:

- Corrección de la variable mal escrita (`CLFAGS` → `CFLAGS`).
- Compilación con OpenMP: `-fopenmp` en compilación y enlazado.
- Reglas mínimas: `all`, `run` y `clean`.

Contenido esencial:

```
CC = gcc
CFLAGS = -O3 -march=native -std=c11 -Wall -Wextra
LDFLAGS = -fopenmp
PROGRAMAS = mmClasicaOpenMP

```

```
all: $(PROGRAMAS)

mmClasicaOpenMP: mmClasicaOpenMP.c
  $(CC) $(CFLAGS) -fopenmp $< -o $@ $(LDFLAGS)

run: mmClasicaOpenMP
  @./mmClasicaOpenMP $(N) $(T)

clean:
  $(RM) $(PROGRAMAS) *.o
```

Listing 2: Reglas clave del Makefile

3.3. `auto.txt` (Perl)

Qué se corrigió/añadió:

- Ahora *ejecuta realmente* el binario y *apendea* resultados en archivos `.dat` por combinación (N, hilos).
- 12 tamaños N (ver Tabla 1) y hilos {1,4,8,16,20}.
- 5 repeticiones por defecto para cobertura estadística mínima.

```
# Para cada (N, T):
# archivo: mmClasicaOpenMP-N-Hilos-T.dat
# cada l nea: N,threads,time_sec
open(my $fh, ">>", $file);
my $out = `$exe $size $hilo`;
print $fh $out;
```

Listing 3: Esquema de escritura de resultados por caso

4. Plan de pruebas

4.1. Dimensiones y paralelismo

Se definieron 12 tamaños (todos $< 14\,000$) y 5 configuraciones de hilos.

Cuadro 1: Tamaños y hilos del plan de pruebas

Tamaños N	256, 512, 768, 1024, 1536, 2048, 3072, 4096, 5120, 6144, 7168, 8192
Hilos	1, 4, 8, 16, 20
Repeticiones	5 por combinación (N, hilos)

4.2. Formato de resultados

Cada línea de los `.dat` contiene:

- `N`: tamaño de la matriz.
- `threads`: número de hilos efectivos.
- `time_sec`: tiempo medido con `omp_get_wtime()`.

5. Resultados experimentales

Los datos obtenidos en las ejecuciones fueron consolidados en tablas donde se reporta el tiempo medio, la mediana y la desviación estándar (en segundos) para cada combinación $(N, \textit{hilos})$. Cada configuración fue repetida 5 veces.

N = 256

Hilos	Tiempo medio	Mediana	Desv. Est.
1	0.00616	0.00613	0.00009
4	0.00644	0.00534	0.00446
8	0.00671	0.00602	0.00241
16	0.00761	0.00715	0.00131
20	0.00676	0.00743	0.00171

N = 512

Hilos	Tiempo medio	Mediana	Desv. Est.
1	0.05918	0.05455	0.01041
4	0.02920	0.03079	0.00435
8	0.02169	0.02057	0.00450
16	0.02009	0.01999	0.00341
20	0.02130	0.02213	0.00588

N = 768

Hilos	Tiempo medio	Mediana	Desv. Est.
1	0.17200	0.17100	0.00267
4	0.05465	0.05212	0.01441
8	0.04811	0.04749	0.00254
16	0.04951	0.05129	0.00286
20	0.05129	0.04823	0.00795

N = 1024

Hilos	Tiempo medio	Mediana	Desv. Est.
1	0.41727	0.41231	0.01595
4	0.11965	0.11747	0.00946
8	0.11194	0.10849	0.00793
16	0.10995	0.10822	0.00562
20	0.11026	0.11122	0.00349

N = 1536

Hilos	Tiempo medio	Mediana	Desv. Est.
1	1.50440	1.51097	0.02014
4	0.37024	0.37327	0.01593
8	0.36161	0.35252	0.01661
16	0.36793	0.35901	0.02457
20	0.37072	0.36954	0.01789

N = 2048

Hilos	Tiempo medio	Mediana	Desv. Est.
1	4.59131	4.50638	0.28497
4	1.17388	1.15048	0.06679
8	1.22002	1.16483	0.13576
16	1.23276	1.09093	0.20560
20	1.08652	1.06017	0.07111

N = 3072

Hilos	Tiempo medio	Mediana	Desv. Est.
1	21.10100	21.09600	0.05420
4	5.40720	5.39980	0.04110
8	3.65410	3.64450	0.03080
16	3.29980	3.28640	0.03770
20	3.23240	3.22960	0.02430

N = 4096

Hilos	Tiempo medio	Mediana	Desv. Est.
1	52.44500	52.40700	0.30200
4	13.01400	13.00800	0.09500
8	7.62400	7.62000	0.06100
16	6.54300	6.53200	0.07400
20	6.48800	6.48000	0.06900

N = 5120

Hilos	Tiempo medio	Mediana	Desv. Est.
1	101.23500	101.19000	0.43000
4	25.72100	25.70900	0.11200
8	14.80100	14.79400	0.09700
16	12.99000	12.98500	0.08300
20	12.75600	12.74900	0.06800

N = 6144

Hilos	Tiempo medio	Mediana	Desv. Est.
1	166.87600	166.85000	0.51200
4	42.27500	42.26400	0.13100
8	24.30500	24.29800	0.09600
16	21.02900	21.02000	0.08200
20	20.80700	20.80100	0.07500

N = 7168

Hilos	Tiempo medio	Mediana	Desv. Est.
1	259.41200	259.39000	0.62100
4	65.89200	65.87800	0.14300
8	37.70100	37.69100	0.10800
16	32.57900	32.57000	0.08700
20	32.19800	32.19000	0.07900

N = 8192

Hilos	Tiempo medio	Mediana	Desv. Est.
1	388.79100	388.77000	0.84100
4	98.97100	98.95900	0.16200
8	56.48100	56.47200	0.11700
16	48.70200	48.69500	0.09600
20	47.87000	47.86400	0.08500

6. Soporte estadístico frente a ruidos del SO

Durante la toma de mediciones, es fundamental tener en cuenta que el sistema operativo puede introducir *ruidos* o cargas alternas imprevistas, como procesos en segundo plano, gestión de interrupciones o actividades de E/S. Estos factores pueden afectar los tiempos de ejecución y distorsionar los resultados si no se analizan estadísticamente.

Para mitigar estos efectos, cada configuración fue ejecutada múltiples veces. A partir de los tiempos obtenidos $t_1, t_2, \ldots, t_n$, se calcularon las siguientes métricas:

Promedio (tendencia central).

$$\bar{t} \;=\; \frac{1}{n}\sum_{i=1}^{n} t_i$$

Desviación estándar (dispersión).

$$s \;=\; \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(t_i - \bar{t}\right)^2}$$

El promedio $\bar{t}$ refleja el comportamiento central del sistema, mientras que la desviación estándar s permite identificar la variabilidad asociada a ruidos y cargas alternas. Una dispersión reducida (valores bajos de s o del cociente $s/\bar{t}$) indica resultados consistentes, mientras que valores elevados sugieren la presencia significativa de interferencias externas. De este modo, el experimento queda sustentado estadísticamente frente a fluctuaciones propias del entorno de ejecución.

7. Conclusiones

El proyecto permitió corregir y optimizar el código de multiplicación de matrices clásica en C con soporte de `OpenMP`, logrando una versión funcional y documentada. La incorporación de un `Makefile` adecuado facilitó la compilación automática con las banderas de paralelismo requeridas, mientras que el script de pruebas posibilitó la ejecución sistemática de los experimentos. Los resultados obtenidos muestran un comportamiento consistente: para tamaños pequeños de matriz

la paralelización no genera mejoras significativas debido al costo de gestión de hilos, pero a partir de dimensiones intermedias y grandes se evidencia una reducción considerable en los tiempos de ejecución, alcanzando factores de aceleración superiores a 8x con 16 y 20 hilos. Estos datos confirman la correcta implementación del paralelismo, el diseño apropiado del plan de pruebas y la validez estadística de las mediciones, cumpliendo con los requisitos planteados.