

# Computación de Alto Desempeño

2023-1 - Clase 2954

Evaluación del Rendimiento

Néstor José Aparicio Huertas Oscar Manuel Beltrán Camacho Gisell Natalia Cristiano Muñoz Elkin Daniel Prada Gómez

- nestorj\_aparicioh@javeriana.edu.co
- beltranom@javeriana.edu.co
- gn cristiano@javeriana.edu.co
- elkind.prada@javeriana.edu.co

### **AGENDA**



### 1. Introducción:

- Computación paralela y HPC.
- OpenMP y MPI.
- Objetivo del estudio.

### 2. Background:

Conceptos Claves

### 3. Configuración de Experimentos:

- Esquema variables experimentales.
- Entrada Experimentos Lanzador.
- MM1c, MM1f, MM1fu y MM2f.

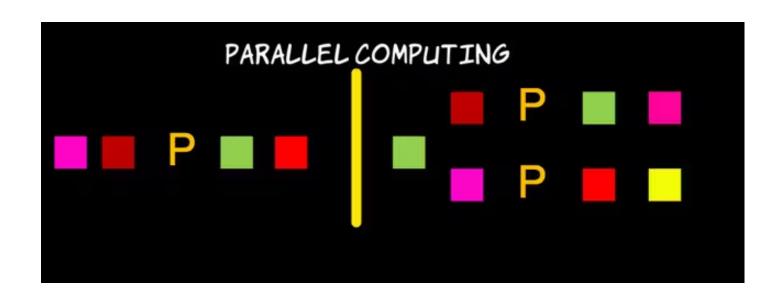
### 4. Resultados y análisis

- OpenMP
- MPI
- OpenMP vs MPI

### 5. Conclusiones

### Introducción Computación Paralela y HPC





**Principio básico:** El problema se puede dividir en subproblemas, cada uno de los cuales se puede resolver simultáneamente.

## Niveles de Paralelismo





Nivel de tarea.



La computación paralela es una estrategia clave empleada en la computación de alto rendimiento.



API's con las cuales se puede construir programas en paralelo:

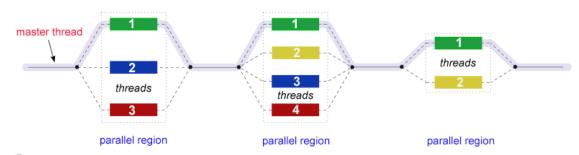
**OpenMP y MPI** 



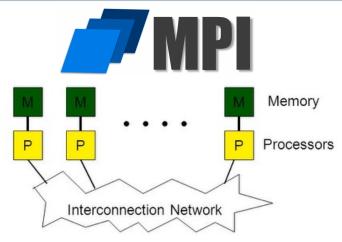
### Introducción OpenMP y MPI



## **OpenMP**



- OpenMP permite la ejecución fork-and-join, donde un programa se inicia como un único proceso o subproceso.
- Durante la ejecución, el subproceso se ejecuta secuencialmente hasta que se encuentra una directiva de paralelización
- En ese momento, se crean grupos de subprocesos y el subproceso actual se convierte en el subproceso maestro, y todos los hilos ejecutan el programa hasta el final de la región paralela



- Además de la paralelización dentro de un nodo, se puede lograr otra paralelización mediante la programación de la interfaz de paso de mensajes (MPI).
- MPI es un estándar para escribir programas de paso de mensajes que permite la comunicación entre procesos llamando a rutinas de la biblioteca para enviar y recibir mensajes.
- Esta combinación de técnicas de programación ofrece una gran capacidad de paralelización y escalabilidad en sistemas de alto performance.

### Introducción Objetivo del Estudio





Analizar y comparar el rendimiento de cuatro algoritmos de multiplicación de matrices estándar con diferentes técnicas de optimización.



Implementar los algoritmos con dos modelos de programación paralela: MPI y OpenMP.



Evaluar el tiempo de ejecución de la multiplicación de matrices para diferentes tamaños y número de hilos en una arquitectura de memoria distribuida y compartida.



Evaluar el tiempo de ejecución de la multiplicación de matrices para diferentes tamaños y número de hilos en una arquitectura de memoria compartida.

### Background **Conceptos Claves**



### **CPU**

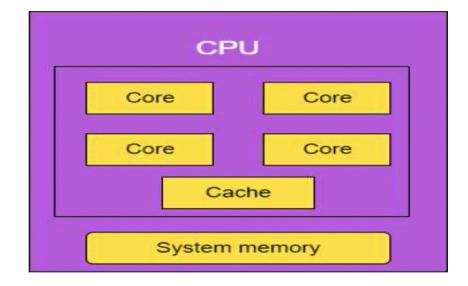
La Unidad Central de Procesamiento (CPU) es el circuito electrónico ejecuta las que instrucciones que componen un programa de computadora.



### CORE

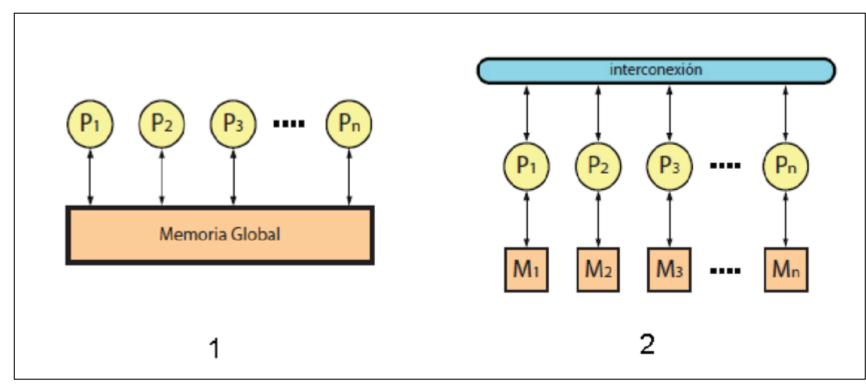
Un core es el cerebro de la CPU recibe que instrucciones y realiza operaciones. Puede tener multi-cores, por ejemplo, puede tener 2 cores, 4 cores, 8 cores, etc.





## Background Conceptos Claves





Modelos de programación paralela basados en memoria compartida:

- Modelos de subprocesos.
- Modelos basados en directivas.
- Modelos de asignación de tareas.

Diferencias entre los modelos (1) memoria compartida y (2) memoria distribuida.

Los modelos empleados son: OpenMP como modelo de memoria compartida basado en directivas y MPI como modelo de memoria distribuida y compartida.

## Configuración de Experimentos Esquema Variables Experimentales



### **Variables Independientes**

#### Tamaño de las Matrices:

- Aumentando el tamaño de la matriz de 100 a 1000 usando el paso 100: 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000.
- Aumentando el tamaño de la matriz de 1200 a 3000 usando el paso 200: 1200, 1400, 1600, 1800, 2000, 2400, 2800, 3000.

#### Número de hilos:

2,4,6,8,10,12,14,16,18,20

#### **Factores NO Controlables**

- Características y capacidades del sistema informático utilizado.
- Eficiencia y rendimiento de las implementaciones de los algoritmos de multiplicación de matrices.
- Otros factores externos que podrían influir en el tiempo de ejecución, como la carga de trabajo del sistema operativo.



Sistema de Unidad Experimental (180 Escenarios)



#### **Factores Controlables**

- Tamaño de las matrices a emplear en cada experimento.
- Cantidad de hilos a emplear en cada experimento.
- Configuración y ejecución del algoritmo de multiplicación de matrices.

#### **Variables Dependientes**

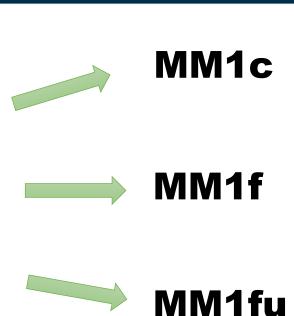
Tiempo de ejecución en segundos.

### Configuración de Experimentos Entrada Experimentos



Sistema de Unidad Experimental (180 Escenarios)

¡Cada escenario de configuración se repitió 30 veces!





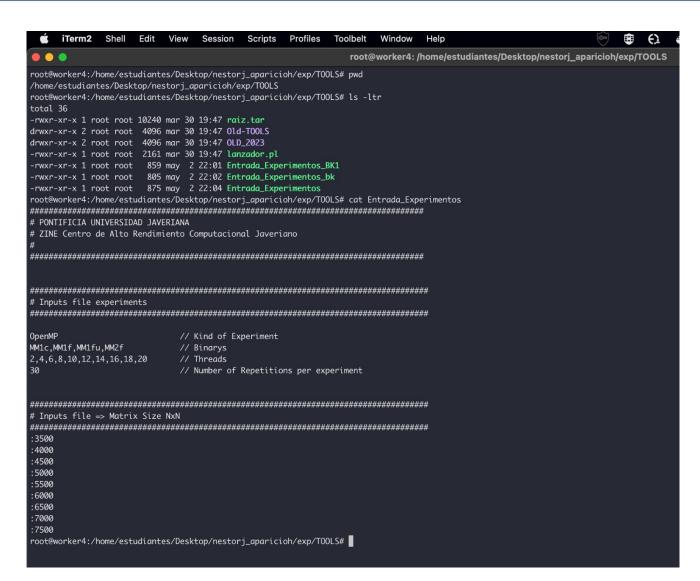




### Configuración de Experimentos Entrada Experimentos OpenMP



Ejemplo de cómo se debe configurar el archivo para el ejercicio propuesto (OpenMP):



### Configuración de Experimentos Lanzador OpenMP



Este archivo se encarga de ejecutar el ciclo de procesos configurados en el archivo anterior.

> El lanzador se ejecuta con el comando:

```
./lanzador.pl <archivo_con_parametros>
./lanzador.pl Entrada_Experimentos
```

➤ Para las ejecuciones largas como la propuesta en el ejercicio, se sugiere utilizar el nohup para que la ejecución se haga en segundo plano:

```
nohup ./lanzador.pl Entrada Experimentos &
```

```
View Session Scripts Profiles
                                                                  root@worker4: /home/estudiantes/Desktop/nestori_aparicioh/exp/TOOLS
 root@worker4:/home/estudiantes/Desktop/nestorj_aparicioh/exp/T00LS# pwd
/home/estudiantes/Desktop/nestori_aparicioh/exp/TOOLS
root@worker4:/home/estudiantes/Desktop/nestorj_aparicioh/exp/TOOLS# ls -ltr
 -rwxr-xr-x 1 root root 10240 mar 30 19:47 raiz.tar
 drwxr-xr-x 2 root root 4096 mar 30 19:47 Old-TOOLS
 root root 2161 mar 30 19:47 lanzador.pl
                        859 may 2 22:01 Entrada_Experimentos_BK1
 rwxr-xr-x 1 root root 805 may 2 22:02 Entrada_Experimentos_bk
 -rwxr-xr-x 1 root root 875 may 2 22:04 Entrada_Experimentos
 oot@worker4:/home/estudiantes/Desktop/nestorj_aparicioh/exp/T00LS# cat lanzador.pl
# Created by: John Corredor UAB-CAOS
 john@aopcjc.uab.es
# Julio 2008
# Lanza el binary juntos con los argumentos desde argumentos.txt
# y los eventos, desde el archivo Entrada_Experimentos
 if (@ARGV[0]) {
        FInput = "ARGV[0]";
 else {
        usage();
 use Switch;
path0 = pwd;
chomp($path0);
T = index(\frac{path0}{T})-1;
$Path = substr($path0,0,$T);
@Labels = ("Experiment", "Binarys", "Threads", "Repetitions");
&Search(\@Labels):
               = \frac{1}{2}
$Experiment
               = split(/,/,$Inputs[1]);
@Binarys
               = split(/,/,$Inputs[2]);
@Threads
               = \frac{1}{2}
@Separate = ("-",":");
```

### Configuración de Experimentos Entrada Experimentos MPI



Ejemplo de cómo se debe configurar el archivo para el ejercicio propuesto (MPI):

```
drwxrwxr-x 2 sistemas sistemas 12288 may 10 21:29 MM1c
sistemas@manager2:/nfs/condor/nestorj_aparicioh$ cat parametrosMPI.txt
threads=(2 4 6 8 10 12 14)
matrixes=(100 200 300 400 500 600 700 800 900 1000 1200 1400 1600 1800 2000 2400 2800 3000 3500 4000 4500 5000 5500 6000 6500 7000 7500)
programs=("MM1c")
sistemas@manager2:/nfs/condor/nestorj_aparicioh$
```

### Configuración de Experimentos Lanzador MPI



Este archivo se encarga de ejecutar el ciclo de procesos configurados en el archivo anterior.

> El lanzador se ejecuta con el comando: nohup ./lanzadorMPI.sh &

```
• • •
                                                                             sistemas@manager2: /nfs/condor/nestorj_aparicioh
 1 //bin/bash
 3 # Execution Parameters
 4 source parametrosMPI.txt
 6 # Iteration executions
 7 for ((i=0; i<10; i++));
 9 echo "Main FOR - Execute i = ${i}"
      for thread in "${threads[@]}";
       echo "Thread - Execute Thread = ${thread}"
           for matrix in "${matrixes[@]}";
           echo "Matrix - Execute matrix = ${matrix} thread = ${thread}"
               for program in "${programs[@]}";
               filename="MPI-${program}-${matrix}-TH-${thread}.txt"
               echo "Execute MPI-${program}-${matrix}-TH-${thread}"
               mpirun --mca btl_tcp_if_include eno1 --host worker4:${thread},worker5:${thread} mpi${program} ${matrix} >> ${program}/${filename}
               #Comando de referencia:
               #Se debe incluir el parametro --mca btl_tcp_if_include eno1 para
               #restringir el socket de red que se puede usar, pues el proceso
               #estaba evaluando todos, incluido el de docker instalado
               #recientemente.
               #mpirun --mca btl_tcp_if_include eno1 --host worker4:2,worker5:2 ./mpiMM1c 300
               #echo "Ejecutando $cmd"
               #$cmd > "salida_${program}_${thread}_${matrix}.txt"
35 echo "Todas las ejecuciones han finalizado."
```

### Configuración de Experimentos Entrada Experimentos MPI



**Nota**: Para la ejecución del experimento con MPI utilizaremos recursos distribuidos (memoria compartida) entre los nodos del cluster worker 4 y worker5, como lo indica el comando del archivo lanzadorMPI.sh visto anteriormente:

```
mpirun --mca btl_tcp_if_include eno1 --host
worker4:${thread},worker5:${thread} mpi${program} ${matrix}
>> ${program}/${filename}
```

### Configuración de Experimentos Lanzador MPI



manager2: /nfs/d	ondor/nest	orj_aparicio	h/MM2f (s	ssh)	<b>¥1</b>		sistemas@n	nanager2: ~ (	(ssh)	₩2	sistemas@worker4: ~ (ssh
top - 08:49:16 u	o 8 days,	14:10,	2 user	s, lo	ad aver	ige: 10,5	2, 5,07,	3,36			
Tasks: 398 total	, 17 rur	ıning, 38	<b>31</b> sleep		0 stopp		zombie				
%Cpu0 : <b>100,0</b> us			ni, <b>0,</b> 0		<b>0,0</b> wa,						
%Сри1 : <b>99,7</b> us					<b>0,0</b> wa,			<b>0,0</b> st			
%Cpu2 : <b>6,3</b> us			ni, <b>93,4</b>		<b>0,0</b> wa,	<b>0,0</b> hi,					
%Cpu3 : <b>100,0</b> us					<b>0,0</b> wa,	<b>0,0</b> hi,		<b>0,0</b> st			
%Cpu4 : <b>9,6</b> us			ni, <b>89,4</b>		<b>0,0</b> wa,	0,0 hi,					
%Cpu5 : <b>100,0</b> us					<b>0,0</b> wa,	0,0 hi,					
%Cpu6 : <b>69,3</b> us			ni, 30,3		0,0 wa,						
%Cpu7 : <b>100,0</b> us					0,0 wa,	0,0 hi,					
%Cpu8 : <b>99,7</b> us					<b>0,0</b> wa,	0,0 hi,					
%Cpu9 : <b>100,0</b> us					<b>0,0</b> wa,	0,0 hi,					
%Cpu10 : <b>100,0</b> us					<b>0,0</b> wa,	0,0 hi,					
%Cpu11 : <b>100,0</b> us %Cpu12 : <b>95,0</b> us					<b>0,0</b> wa, <b>0,0</b> wa,	0,0 hi, 0,0 hi,					
%Сри12 : <b>93,0</b> us %Сри13 : <b>99,7</b> us					<b>0,0</b> wa, <b>0,0</b> wa,	0,0 hi,					
%Сри13 : <b>93,7</b> us			ni, <b>5,</b> 0		0,0 wa,	0,0 hi,					
%Cpu15 : <b>100,0</b> us					0,0 wa,	0,0 hi,					
6Cpu16 : <b>31,1</b> us			ni, 68,5		<b>0,0</b> wa,	0,0 hi,					
6Cpu17 : <b>99,7</b> us					0,0 wa,	0,0 hi,					
6Cpu18 : <b>0,7</b> us			ni, <b>98,</b> 3		0,7 wa,						
6Cpu19 : <b>100,0</b> us							0,0 si,				
MiB Mem : 64008			4 free,				,8 buff/c				
	,0 total,		0 free,		<b>0,0</b> used		,9 avail I				
PID USER	PR NI	VIRT	RES	SHR	S %CPU	J %MEM	TIME+	COMMAND			1,22
1220724 sistemas		1829264			R 100,3		0:05.66				
L220733 sistemas		1829264	95508		R 100,3		0:05.65 i				
L220735 sistemas		1829264	93556		R 100,3		0:05.66 i				
1220815 sistemas		1758980	98560		R 100,3		0:03.66				
L220719 sistemas		1829408			R 100,0		0:05.65				
L220720 sistemas		1829264	95524		R 100,0		0:05.65				
1220721 sistemas		1829264	94228		R 100,0		0:05.65				
1220722 sistemas		1829264	94240		R 100,0		0:05.65				
1220723 sistemas		1829264	94296		R 100,0		0:05.65				
1220726 sistemas		1829264	94236		R 100,0		0:05.65				
.220730 sistemas		1829264	95620		R 100,0		0:05.65				
.220737 sistemas		1829264			R 100,0		0:05.65				
1220739 sistemas		1829264	94148		R 100,0		0:05.65				
L220812 sistemas		1758988			R 100,0		0:03.66				
		1758980	98532		R 100,0		0:03.66				
1220813 sistemas						T 1/2					
122 <b>0813 sistemas</b> 1 <b>220814 sistemas</b> 1152 avahi	<b>20 0</b> 20 0	<b>1758980</b> 13992	<b>98524</b> 9504	3428	R 100,0		ا <b>0:03.65</b> 23:15.93				

## Configuración de Experimentos MM1c, MM1f, MM1fu y MM2f.



Los programas difieren en la forma como se almacenan las matrices en memoria, lo que afecta el orden en que se acceden a las entradas:



#### MM1c

- Programa en C.
- Enfoque estándar de bucles anidados triple.
- Accede a los elementos de B en orden de columna principal.



#### MM1f

- Programa en C.
- Enfoque estándar de bucles anidados triple.
- Accede a los elementos de B en orden de fila principal.



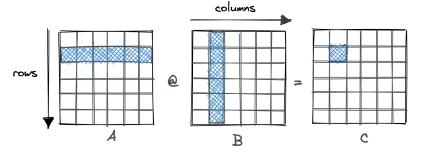
#### MM1fu

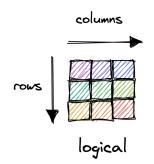
- Programa en C.
- Enfoque estándar de bucles anidados triple.
- Se agregó un bucle adicional en el que se realizan los cálculos en bloques de cuatro elementos y se suman los resultados de cada bloque.

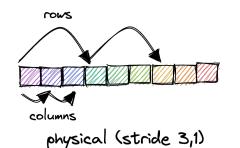


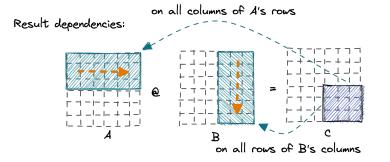
#### MM2f

- Programa en C.
- Multiplicación de matrices por bloques.
- Las matrices de entrada se dividen en bloques de tamaño 2x2.











Para garantizar que el proceso de evaluación comparativa sea consistente, cada escenario se ejecutó 30 veces y se calculó el tiempo promedio de ejecución.

Average of Seconds	Etiquetas de columna	1																		
Etiquetas de fila		100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	30001	otal general
2	0,000346	167 0,0	003128967	0,010976233	0,0270999	0,052482714	0,0960847	0,1438056	0,2218138	0,305484933	0,527339286	0,80361616	1,439023556	4,349493483	8,437843033	12,8241129	25,28362777	45,3863545	56,44949383	8,652640094
4	0,000179	233 0,0	001596733	0,0055602	0,013946667	0,026396914	0,046628233	0,0720951	0,115815367	0,153770567	0,268720171	0,4080311	0,717562533	2,246682733	4,2358426	6,477830767	12,78402807	22,811205	29,1330067	4,340077356
6	0,000120	967 0,0	001101633	0,003760133	0,0093828	0,0179862	0,031731067	0,050962467	0,0833249	0,110667633	0,1859664	0,2883178	0,503144433	1,606454133	2,891511933	4,454666433	8,823794933	15,77603317	20,30366877	3,063477544
8	9,403338	-05 0,0	000834267	0,003032367	0,007819567	0,015057171	0,0273863	0,046323267	0,073932533	0,090525867	0,152357743	0,2451518	0,411621	1,300408367	2,303345333	3,532601433	7,072430033	12,74444577	16,32130317	2,420540318
10	9,543338	-05 0,0	000784467	0,0028583	0,007862833	0,015718886	0,027278767	0,053693633	0,141096633	0,1792428	0,239844429	0,282090967	0,4333418	1,207109333	2,017941067	3,087349067	6,129414267	11,10667963	14,2025213	2,1369555
12	0,000	082	0,00065340	0,002483533	0,007379767	0,014205267	0,023992067	0,042578967	0,138168133	0,2146815	0,350497967	0,458528267	0,657113667	1,504479133	2,494097167	3,664927033	6,958963067	12,34956527	15,59637537	2,471042865
14	7,243338	-05 0,0	000573233	0,002112	0,0065858	0,0125147	0,0210843	0,038437833	0,123237933	0,202510667	0,344673267	0,584268233	0,650731433	1,4355505	2,304349667	3,371408967	6,284184867	10,9517146	13,92626567	2,236682006
16	6,546678	-05 0,0	000499533	0,001937533	0,0056255	0,011058267	0,019388	0,0348843	0,1130365	0,190987533	0,330824333	0,6146646	0,840726967	1,498126567	2,325051633	3,343930367	6,174585033	10,6634582	13,62769817	2,210919361
18	6,003338	-05 0,0	000472467	0,0018896	0,005200633	0,009954967	0,017516833	0,0322974	0,104920867	0,174749567	0,311814367	0,634720667	0,9783894	1,5683881	2,295336233	3,263095467	5,883987267	10,064104	12,90285843	2,124986461
20	0,003960	067 0,0	003762433	0,0039597	0,01091	0,0186292	0,021398567	0,028349967	0,085155533	0,153491267	0,282272067	0,591908067	0,9661068	1,5151067	2,2153801	3,177853667	5,796223633	9,802522233	12,65054313	2,073751841
Total general	0,000507	583 0,0	001340713	0,00385696	0,010181347	0,019901272	0,033248883	0,054342853	0,12005022	0,177611233	0,299283153	0,485833386	0,752915074	1,814730696	3,152069877	4,71977761	9,119123893	16,16560824	20,51137345	3,172971456

### Tabla para MM1c

Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	3000	Total general
2	0,000285933	0,003043933	0,010218533	0,0255654	0,049188857	0,0858542	0,138476567	0,206805967	0,2936704	0,402125143	0,6988266	1,144414133	1,775266	2,590798133	3,5705266	6,201278767	9,8684743	12,20535393	2,146112313
4	0,000143467	0,0014747	0,005178033	0,0126224	0,025730257	0,043536633	0,0694338	0,103845633	0,148649467	0,204921514	0,357478367	0,586263633	0,905204767	1,321728767	1,827647967	3,1713979	5,0251974	6,196355967	1,093377416
6	9,82333E-05	0,001018167	0,003487567	0,008548567	0,017055467	0,0296546	0,0476133	0,071306533	0,101147933	0,1396089	0,2443702	0,398940667	0,6155933	0,9050166	1,2536493	2,1691858	3,445024867	4,237547467	0,760492637
8	7,72667E-05	0,000761667	0,002662133	0,006417633	0,013119971	0,0226942	0,036071867	0,0548086	0,081022833	0,112224171	0,193794867	0,318116633	0,4879978	0,712944367	0,9882116	1,712648533	2,719409167	3,3509153	0,590988507
10	8,15333E-05	0,000696567	0,002460867	0,006010833	0,011711286	0,0209482	0,033336533	0,048418833	0,068582933	0,094218057	0,163029	0,266422267	0,4124684	0,6039189	0,8332087	1,446725833	2,3079526	2,844256467	0,500841965
12	0,0000756	0,000596767	0,002060633	0,0051537	0,0101512	0,017453033	0,0289254	0,041603133	0,059331033	0,082306767	0,1423505	0,2282528	0,3509622	0,511367533	0,706104467	1,221404533	1,947208233	2,395316333	0,430590215
14	6,94333E-05	0,000532867	0,0018197	0,004495267	0,0087207	0,015283033	0,0241646	0,036972867	0,0521733	0,071572933	0,124482467	0,197146833	0,302938767	0,445270167	0,6124943	1,0633185	1,687892433	2,0875922	0,374274465
16	5,94333E-05	0,000454267	0,001576933	0,0038555	0,007833867	0,013668167	0,021716367	0,032509067	0,0473395	0,064801	0,109550367	0,175794267	0,268294433	0,3929451	0,5384989	0,9372532	1,495476767	1,843485467	0,330839589
18	0,0000632	0,0004145	0,001325333	0,0031925	0,0069207	0,012425467	0,019644567	0,029434	0,041907867	0,057570367	0,099457567	0,158261	0,243803233	0,352447467	0,4873987	0,852637933	1,359124267	1,665595233	0,299534661
20	0,006634633	0,005458433	0,007512267	0,008673867	0,0144088	0,0212151	0,022230833	0,0301473	0,044452167	0,057853833	0,097668533	0,152209	0,235079033	0,350546533	0,4798915	0,836135767	1,3303315	1,652117167	0,297364793
Total general	0,000758873	0,001445187	0,0038302	0,008453567	0,017012453	0,028273263	0,044161383	0,065585193	0,093827743	0,133386016	0,223100847	0,362582123	0,559760793	0,818698357	1,129763203	1,961198677	3,118609153	3,847853553	0,685385688



Para garantizar que el proceso de evaluación comparativa sea consistente, cada escenario se ejecutó 30 veces y se calculó el tiempo promedio de ejecución.

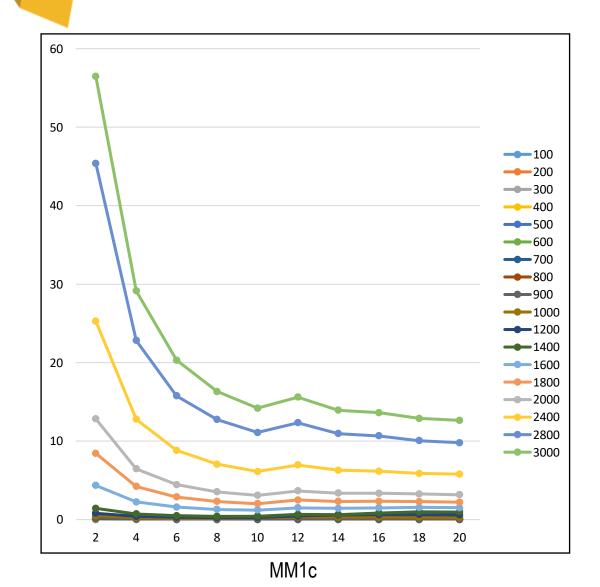
Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	3000	Total general
2	0,000265117	0,002184567	0,007217933	0,018150333	0,0352715	0,059664033	0,09823905	0,136828283	0,19490335	0,267494767	0,475081133	0,815796933	1,3020244	1,930454767	2,693823267	4,7135544	7,4782428	9,1858639	1,109917952
4	0,00013635	0,001104517	0,00369045	0,008787533	0,01707785	0,0297408	0,046867367	0,069873083	0,099215217	0,136277133	0,237941767	0,400199033	0,655975233	0,987195533	1,3706941	2,390118867	3,793650133	4,649334633	0,562013806
6	0,00009335	0,00076605	0,002522433	0,006025233	0,01180945	0,020429	0,033630883	0,04902945	0,068177783	0,097022933	0,164485933	0,282561633	0,456546133	0,684747933	0,955070733	1,6734015	2,665026233	3,276648767	0,394091817
8	7,38333E-05	0,00056955	0,001909367	0,00448655	0,00887755	0,015549333	0,02700525	0,040119067	0,055179933	0,078411067	0,135872633	0,2273147	0,3626025	0,5423365	0,758195167	1,3351	2,116826733	2,6098165	0,313852469
10	0,00011025	0,000864783	0,002980583	0,007319833	0,0141854	0,024285517	0,038268617	0,05611705	0,0809095	0,108043267	0,152314533	0,2343815	0,3614781	0,533176033	0,691798133	1,166483233	1,833762	2,25123	0,288249995
12	9,27167E-05	0,000734567	0,002433417	0,005936367	0,011927017	0,02088545	0,033359567	0,052627583	0,0697815	0,0983761	0,1706013	0,272670033	0,416915433	0,6018368	0,803490467	1,351547967	2,132478367	2,587292433	0,32706538
14	8,23167E-05	0,00064155	0,002098567	0,005073267	0,010438217	0,018203117	0,029128267	0,0463212	0,061813417	0,086496133	0,1491757	0,237745	0,365577467	0,527049667	0,718818567	1,218671233	1,902419467	2,3371393	0,292247865
16	7,41833E-05	0,000564017	0,001951783	0,00487045	0,009684517	0,016895933	0,02715235	0,040366317	0,056083067	0,077751267	0,133825867	0,213977133	0,327671333	0,478057567	0,662632467	1,163965267	1,846575367	2,265541267	0,277232695
18	7,01833E-05	0,000529633	0,001794567	0,00441145	0,008665517	0,015316483	0,024111367	0,03644395	0,051276833	0,070908667	0,122780333	0,195147733	0,2986544	0,4368827	0,6072977	1,061492233	1,7031926	2,101327933	0,254923121
20	0,005767917	0,005541633	0,0080436	0,00787015	0,011368067	0,011724767	0,024301083	0,030757683	0,04359445	0,0569236	0,101770367	0,172948433	0,278271867	0,408252867	0,5651762	0,992984567	1,571080233	1,933968433	0,236270936
Total general	0,000676622	0,001350087	0,00346427	0,007293117	0,013930508	0,023269443	0,03820638	0,055848367	0,078093505	0,107770493	0,184384957	0,305274213	0,482571687	0,712999037	0,98269968	1,706731927	2,704325393	3,319816317	0,405586604

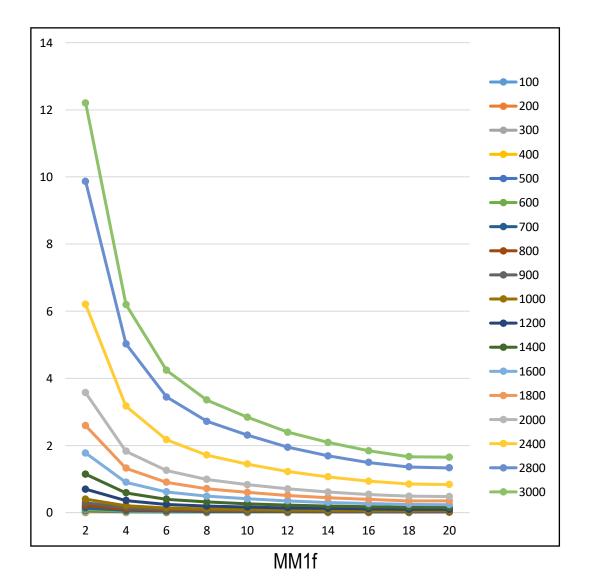
Tabla para MM1fu

Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	30001	Total general
2	0,0001019	0,000830283	0,002610733	0,006254783	0,012749867	0,024143917	0,0356248	0,051097117	0,072081067	0,098463267	0,1727196	0,3032825	0,490103533	0,741069967	1,059327633	1,845118833	2,877548567	3,673580633	0,432303832
4	5,29833E-05	0,000427183	0,00136015	0,003167267	0,006110833	0,0106203	0,01652415	0,02600945	0,036162667	0,050357067	0,089930533	0,153205267	0,240282233	0,366105367	0,527019033	0,917572667	1,450806867	1,820447433	0,215429498
6	0,00003855	0,000293767	0,00092205	0,002216883	0,004297933	0,007712617	0,012073117	0,018974883	0,026909667	0,037660733	0,0666038	0,1104271	0,172193	0,259324133	0,3735832	0,649544433	1,026649333	1,289430967	0,153047986
8	0,00003275	0,000229433	0,000706633	0,001677883	0,0032488	0,006079883	0,0109058	0,015640117	0,024747467	0,032378667	0,058601967	0,0894978	0,1370736	0,2072851	0,3015866	0,521898333	0,826734767	1,038248467	0,123697883
10	3,79167E-05	0,000275683	0,00092865	0,002341617	0,0047933	0,008391867	0,013436717	0,019974133	0,027971767	0,039139533	0,068253533	0,1033147	0,154121133	0,211685367	0,2971498	0,479691033	0,7448167	0,956166733	0,11891266
12	0,00003695	0,0002441	0,000800533	0,002026	0,004055467	0,0070109	0,01146375	0,01707705	0,0243917	0,033710867	0,058392633	0,090697567	0,146331233	0,213676667	0,3034202	0,525964867	0,819130767	1,022716067	0,124009399
14	0,00003505	0,00022125	0,00071525	0,001778917	0,003480883	0,006129733	0,009838733	0,01479205	0,021927817	0,029017733	0,050714967	0,0795827	0,1249271	0,1883298	0,266997333	0,464578033	0,735727967	0,914629333	0,110086827
16	0,0000336	0,000198083	0,000636167	0,001539217	0,003105317	0,005434383	0,008726167	0,01303375	0,01897935	0,026172367	0,0453177	0,072331367	0,111419167	0,167580667	0,235556733	0,413136933	0,6646688	0,8239056	0,098646719
18	0,0000326	0,000179533	0,000583617	0,0014131	0,00275275	0,004885617	0,007939383	0,011907183	0,016989133	0,023414133	0,041992833	0,0647034	0,103379967	0,152565333	0,211263633	0,3754016	0,605044267	0,754667433	0,089844386
20	0,0047564	0,006890217	0,003747983	0,005982767	0,0067989	0,010518467	0,0149667	0,01721235	0,019779383	0,026744133	0,03147	0,0653288	0,0946065	0,143434	0,2012634	0,3549363	0,5705173	0,706315533	0,087997122
Total general	0,00051587	0,000978953	0,001301177	0,002839843	0,005139405	0,009092768	0,014149932	0,020571808	0,028994002	0,03970585	0,068399757	0,11323712	0,177443747	0,26510564	0,377716757	0,654784303	1,032164533	1,30001082	0,155397631

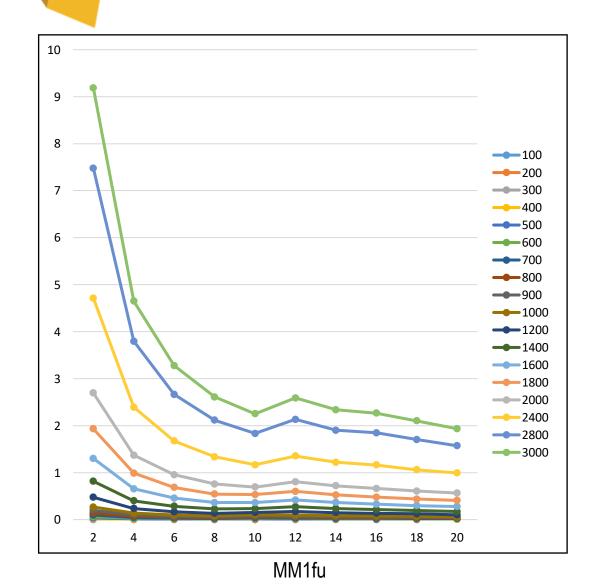
Tabla para MM2f

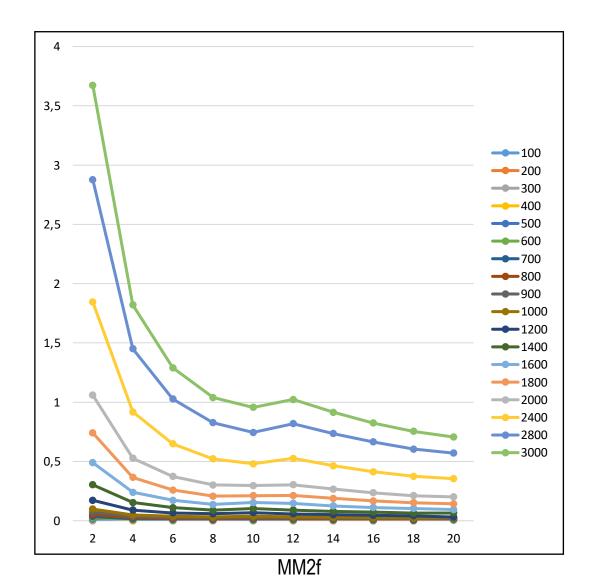




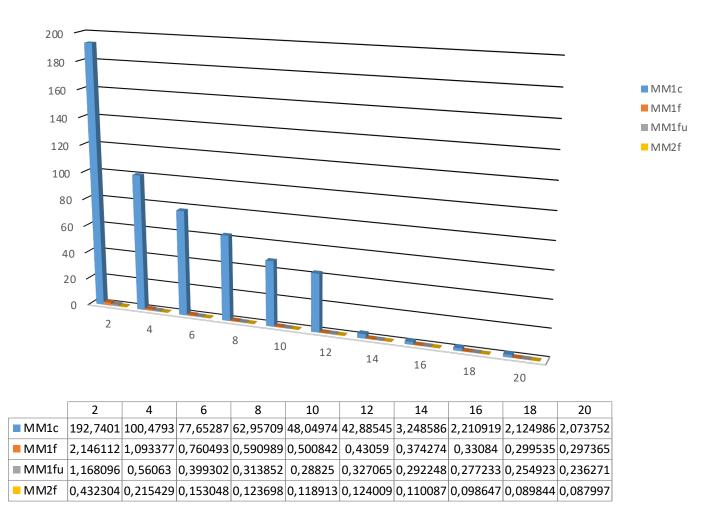












Rendimiento de los algoritmos para diferentes números de threads.



Para garantizar que el proceso de evaluación comparativa sea consistente, cada escenario se ejecutó 30 veces y se calculó el tiempo promedio de ejecución.

Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	3000	Total general
2	0,016692	0,035734857	0,10879	0,131763143	0,274317	0,322559154	0,560503	1,016577643	1,485548357	1,881141071	3,549886643	6,237480571	11,45839214	16,66823321	26,06075336	40,241897	66,42818185	83,83140554	14,03114797
4	0,011513	0,017957077	0,044224692	0,109020692	0,119817077	0,220832385	0,361342538	0,557382385	0,876042615	1,141082538	1,919085462	3,140357154	5,612220615	7,815687692	11,98229831	21,989815	39,30948738	49,64239175	7,869849215
6	0,032687538	0,068460308	0,075975077	0,158627231	0,444901308	0,340416923	0,832528692	0,543048846	0,924875308	1,231360538	1,912758308	2,885723308	5,812599923	7,048345769	9,938028923	18,54166092	28,40917023	32,17140792	6,187365393
8	0,045102231	0,037582692	0,151128769	0,294809385	0,081140923	0,135666538	0,232140769	0,356278462	0,535799846	0,623730769	1,093116308	1,832392077	3,215391385	4,608069769	6,648542154	11,85087992	19,41044315	24,87977431	4,223999415
10	0,025434	0,015963154	0,138836385	0,046606308	0,268675417	0,130518769	0,246017077	0,404153231	0,499836692	0,702939	0,994457231	1,947444538	3,079551846	4,222215	5,428357154	10,67860538	19,07410362	22,54894354	3,928420983
12	0,057056462	0,10864375	0,192558385	0,306498385	0,289732077	0,359956538	0,652971385	1,119570154	1,038524692	0,912188462	1,554867462	2,0209145	3,868609923	4,579415615	6,779008231	12,42838869	20,82149931	24,192106	4,545444276
14	0,068609	0,089107154	0,159261	0,388031769	0,565986077	1,003779231	1,327983538	1,348290308	1,143760538	1,927811	2,073870615	2,206345615	3,204901231	4,823490077	7,489563615	12,93662492	20,33616769	25,49182015	4,810300197
Total general	0,036509967	0,052548659	0,124226696	0,20425438	0,292143407	0,35910422	0,601476457	0,766364033	0,935245576	1,210265598	1,889396011	2,942144143	5,247065913	7,246120451	10,78593398	18,61874164	30,54129332	37,40232942	6,586915694

### Tabla para MM1c

Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	3000	Total general
2	0,019001125	0,057157625	0,069106438	0,133860813	0,258772938	0,446173	0,615924563	0,896292875	1,328708875	1,663959375	3,074816625	4,54109925	6,95644625	7,806059438	11,547865	22,44794181	31,40355675	39,66318925	7,384996222
4	0,027777563	0,035297563	0,08994	0,089542938	0,13209	0,199461357	0,2624905	0,386709438	0,541394938	0,734062938	1,176147813	1,96302425	2,801732625	4,014462688	5,519769438	10,15969288	15,24806263	19,32191507	3,451004758
6	0,022733563	0,02219125	0,054699625	0,124166063	0,179660375	0,2324285	0,295601938	0,382464438	0,593580063	0,624237667	1,332499313	1,973246125	2,568603625	3,754289438	4,595650875	8,120483125	12,63691263	15,88820606	2,974920686
8	0,113020938	0,059805875	0,374383375	0,33895725	0,1762625	0,526229188	0,408432	0,55276375	0,568817313	0,709275313	0,769928938	1,66279375	2,191186938	3,368394563	4,418793063	6,9364694	11,62318313	13,89501713	2,690463279
10	0,125539067	0,17454325	0,333520625	0,62760925	1,083602375	1,40016225	1,383178	1,51971625	1,025905813	0,859065563	1,02418375	1,804139375	2,496040125	3,345883188	4,332650438	6,793514125	10,16390313	12,76061175	2,856915519
12	0,114442438	0,155554875	0,490853438	0,9727565	0,4036315	1,540316438	1,453987063	2,268275375	1,85453025	1,08971975	1,085042563	1,881167	2,539572188	2,652073733	4,0180532	5,235956667	9,744831188	11,92346667	2,700556842
14	0,25125525	0,25008825	0,806403063	1,374015063	2,016538875	2,251896	2,861511438	4,333853	1,791565188	3,953452063	2,507022813	3,313106313	2,90010375	3,8859216	4,451545933	6,192660733	9,010768429	12,67274233	3,517612745
Total general	0,095989009	0,107805527	0,316986652	0,522986839	0,607222652	0,955888591	1,040160786	1,477153589	1,100643205	1,383028153	1,567091688	2,448368009	3,207669357	4,133594173	5,578905891	9,502957917	14,35707226	18,11086466	3,657620079



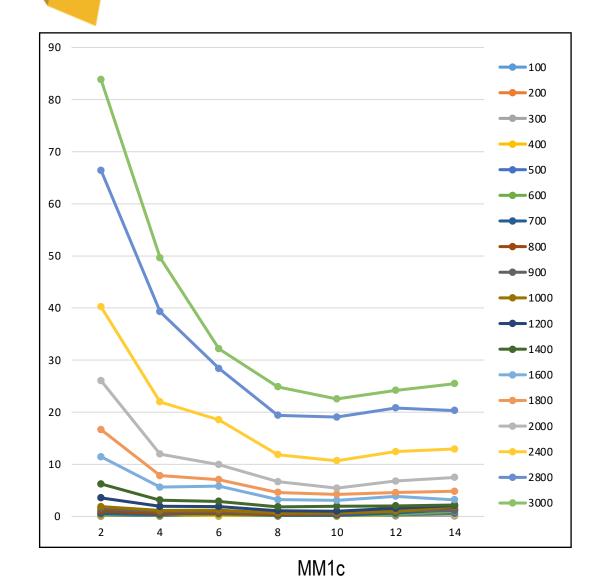
Para garantizar que el proceso de evaluación comparativa sea consistente, cada escenario se ejecutó 30 veces y se calculó el tiempo promedio de ejecución.

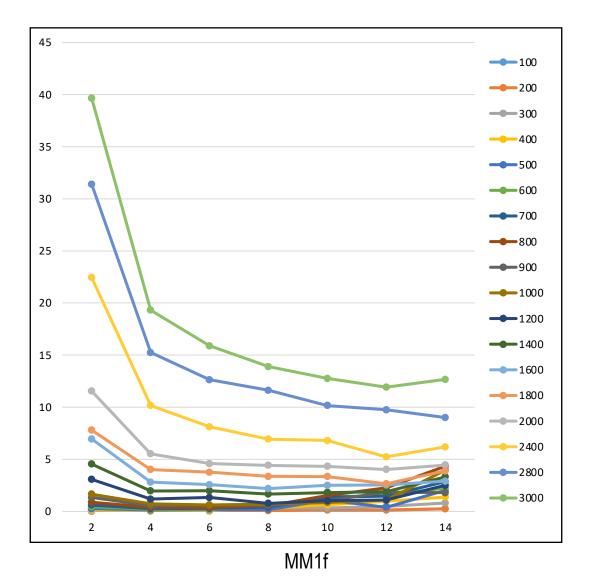
Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	3000	Total general
2	0,015628056	0,022962056	0,038400278	0,081593556	0,139646667	0,198321889	0,337142111	0,501892944	0,703540389	0,892529556	1,495667	2,361091444	3,433069222	5,042189333	6,685276944	11,19508767	18,96123511	22,31506861	4,134463491
4	0,033061333	0,049718278	0,063524	0,091645667	0,128053235	0,164637611	0,256284333	0,315175278	0,497107722	0,667434235	1,098106471	1,532331389	2,014842389	3,463149722	4,429823176	6,229358	9,615375111	12,61082	2,413639519
6	0,081102	0,080770278	0,207198222	0,214785722	0,238087611	0,338453556	0,338531824	0,410611222	0,375053167	0,480266611	0,738572111	1,619743824	2,070437278	2,533373556	3,235105778	5,039373278	8,197248222	9,839602333	2,014486981
8	0,046270944	0,040113556	0,202980167	0,373701778	0,895339611	0,354376444	0,376951944	0,471175167	0,641507222	0,623560667	0,904736778	1,053761444	1,530098167	2,674652056	2,863329444	5,509282	7,868732471	10,32063517	2,023693368
10	0,175769444	0,100160111	0,438297056	0,315632	0,605170389	0,850367556	0,427206389	1,338581889	1,328692333	0,835569722	1,146633706	1,109514389	1,960506	2,135100167	2,976111941	5,547202	7,911477611	8,642308	2,102716357
12	0,059856444	0,078186667	0,398080667	0,751702556	1,134073222	0,354782556	0,715769235	1,119526667	0,679777222	0,616798111	0,949771056	0,983890111	1,6826195	2,625122667	2,830232333	5,169422667	7,668154111	8,516912556	2,022626752
14	0,189596471	0,556471389	0,574638278	1,055114333	1,026893222	1,746465833	2,1815385	2,815248111	2,453741722	1,862205278	1,745254647	2,781797944	2,524780647	2,796876765	3,218661706	5,733327412	8,613145118	9,665866647	2,826777604
Total general	0,08510021	0,132626048	0,274731238	0,412025087	0,599061584	0,572486492	0,664091427	0,996030183	0,954202825	0,855544968	1,149815911	1,634708848	2,170956616	3,04057184	3,753407699	6,351053016	9,859312435	11,71788778	2,505948834

### Tabla para MM1fu

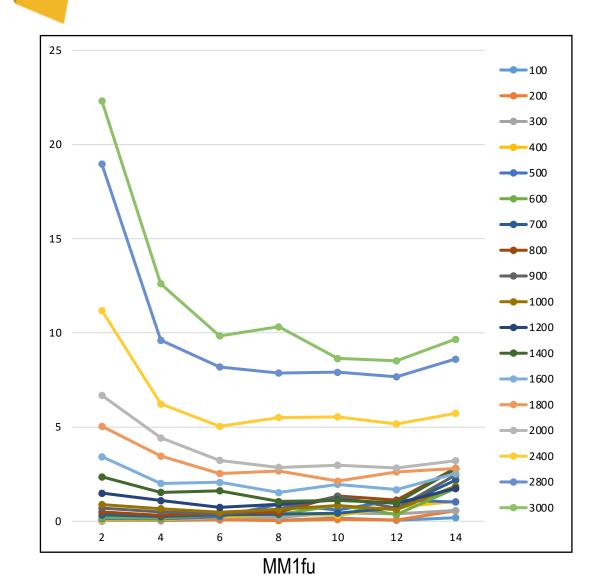
Average of Seconds	Etiquetas de columna																		
Etiquetas de fila	100	200	300	400	500	600	700	800	900	1000	1200	1400	1600	1800	2000	2400	2800	30001	Total general
2	0,01016575	0,024889438	0,0548035	0,115191813	0,19805725	0,346405875	0,534105375	0,815099	1,138313875	1,559033	2,6874075	4,226938375	6,444738438	8,7833435	12,68119844	21,36877513	26,89000706	32,44323181	6,684539174
4	0,028758313	0,029818375	0,041742625	0,1073335	0,188645563	0,250176357	0,357961938	0,515185438	0,715331688	0,858261125	1,440323125	2,143891188	3,036582313	4,247720875	6,406453438	11,10927556	17,22459219	21,27956188	3,913305927
6	0,02521575	0,047532438	0,162018938	0,131499438	0,18304525	0,240343625	0,269353063	0,3736424	0,529889625	0,872835188	1,33378275	1,749178625	2,799416063	3,752756313	5,192388188	8,9345465	13,26534925	15,90153525	3,107510857
8	0,099490563	0,113449667	0,368039125	0,619808938	0,471491	0,332835188	0,75442625	0,949928	0,867745125	0,77691775	1,436283125	2,016049	3,052364813	4,08751675	4,93992925	8,381202625	12,02928338	15,509945	3,175952266
10	0,155082375	0,135148188	0,947498267	0,578375933	1,3162135	0,913434875	1,298821563	1,555630438	1,11033525	1,043152313	1,556805625	2,38844325	2,898243733	3,742516375	4,1846645	6,955758188	9,744034688	13,50016956	3,017421544
12	0,2115085	0,673442938	0,747053875	0,868728875	1,478681563	1,619373625	2,241755125	3,297923938	1,93281325	3,579645875	1,824457875	1,830285188	2,81985275	3,354475938	4,4402475	5,931072688	10,49652788	11,25635788	3,255789181
14	0,1396852	0,2439502	0,410544867	0,721406867	1,398814333	1,124321333	2,460105	1,957544933	1,813473067	2,441064267	1,569232267	2,577692467	2,017932467	2,529892733	2,961578929	5,1430696	6,6125724	8,965762	2,503227007
Total general	0,095304667	0,181220909	0,384992509	0,445252209	0,744444273	0,693629174	1,11895809	1,355527991	1,152368982	1,582463856	1,693724721	2,417495108	3,310817391	4,373348351	5,881638236	9,73005527	13,81608376	17,05170497	3,677994585

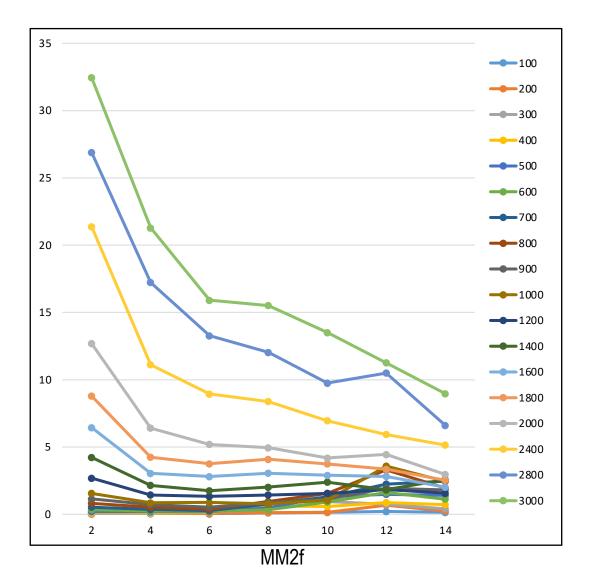




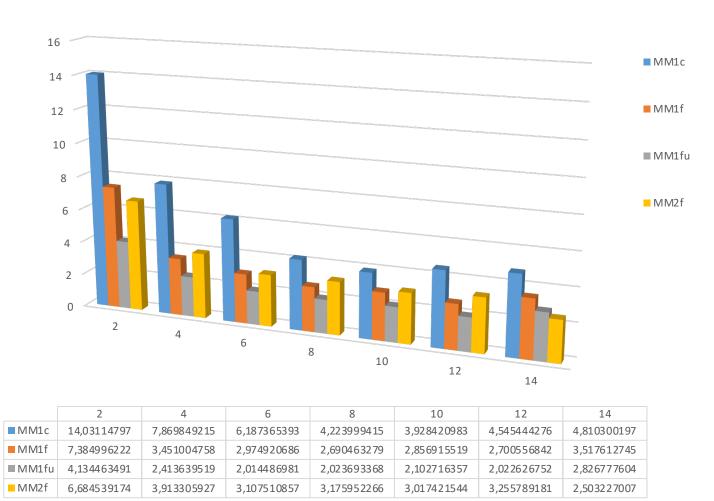






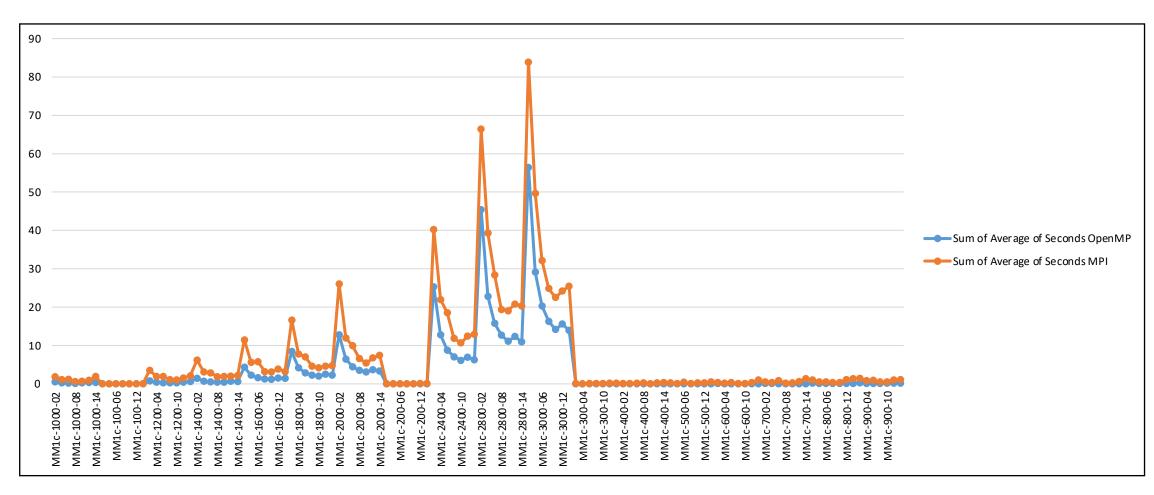




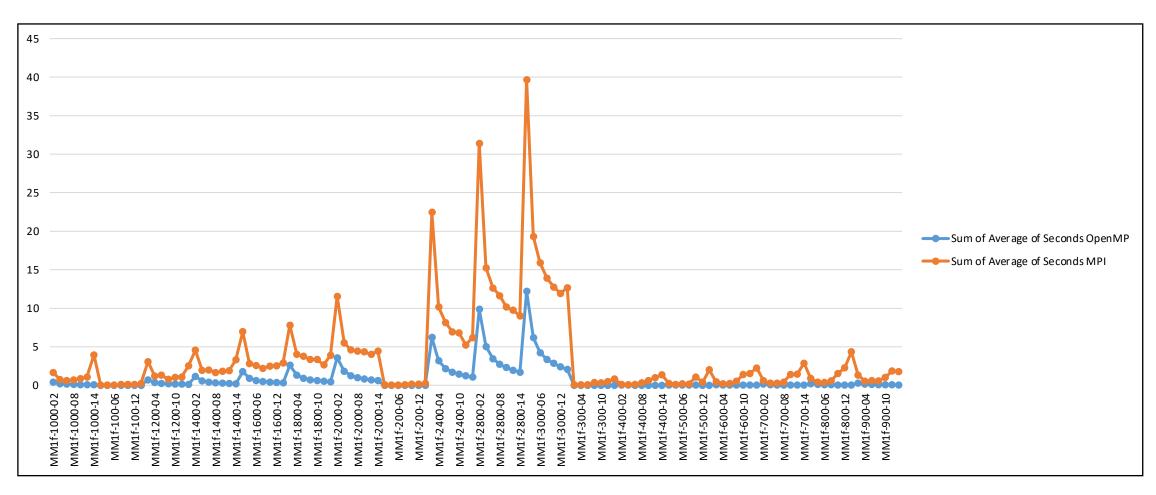


Rendimiento de los algoritmos para diferentes números de threads.

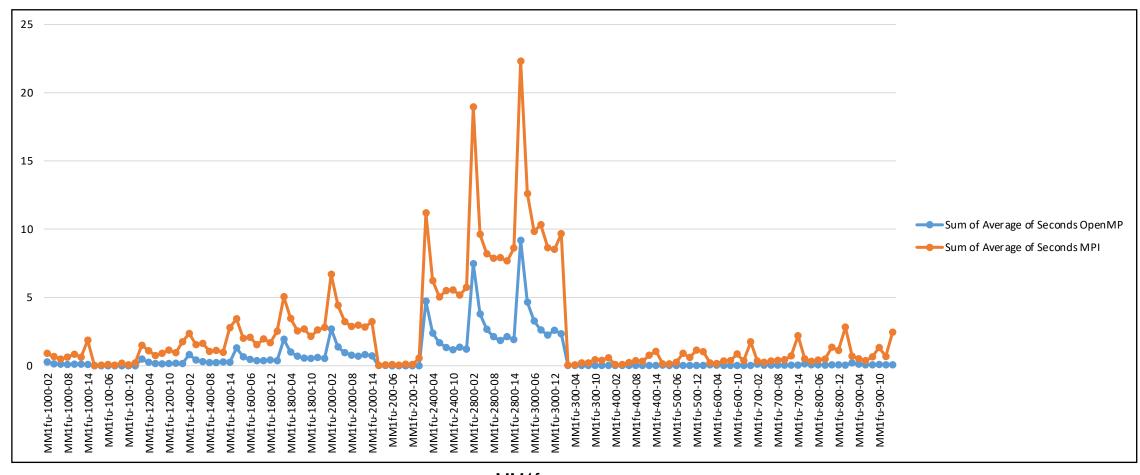




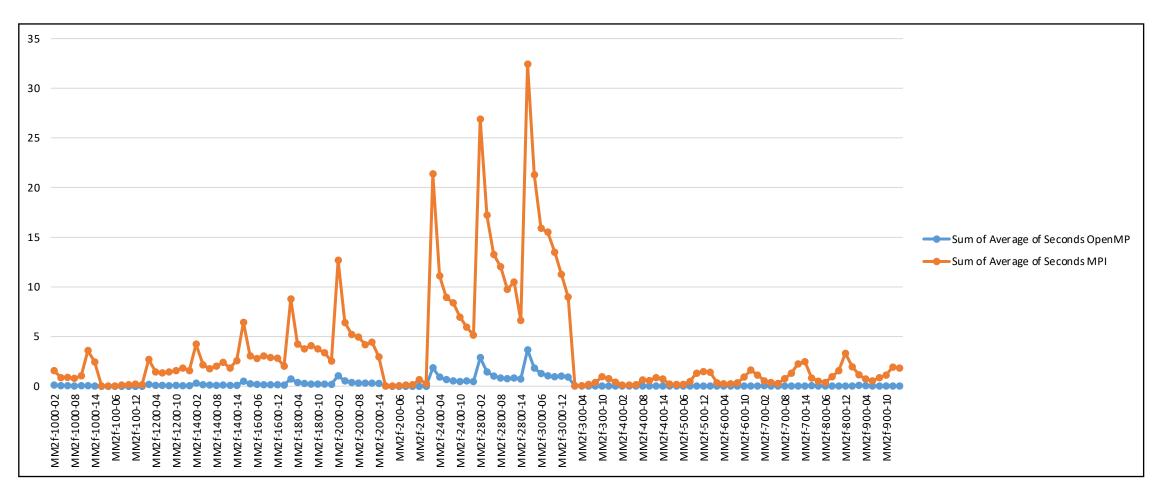












### Conclusiones



- El rendimiento de los algoritmos de multiplicación de matrices mejora al aumentar el número de hilos utilizados, pero se ralentiza a medida que se incrementa, con un límite en el rendimiento alcanzado.
- → MM2f es el código más eficiente en términos de rendimiento, seguido por MM1fu y MM1f. MM1c tiene el rendimiento más bajo entre los cuatro algoritmos.
- ☐ MM2f y MM1fu utilizan un particionamiento de matrices más eficiente y minimizan las operaciones de memoria, lo que mejora su rendimiento y eficiencia en comparación con MM1f y MM1c.
- La elección del mejor modelo para aplicaciones en paralelo depende de varios factores, por un lado la implementación de MPI requiere más esfuerzo y experiencia. Pero, MPI tiene la ventaja de ser compatible con sistemas de memoria compartida y distribuida.
- □ La implementación paralela de algoritmos de multiplicación de matrices mejora el rendimiento. El particionamiento eficiente de matrices y la optimización de operaciones de memoria son clave para obtener un mejor rendimiento.

### Bibliografía



- ➤ Al-Mulhem, M.S., Aidhamin, A., Al-Shaikh, R.: On benchmarking the matrix multiplication algorithm using OpenMP, MPI and CUDA programming languages. ResearchGate (2013)
- ➤ Tim.Lewis: cOMPunity The Independent Community for OpenMP (2019). <a href="https://www.openmp.org/compunity/">https://www.openmp.org/compunity/</a>.
- Nayfeh, B.A., Olukotun, K.: A single-chip multiprocessor. IEEE Computer 30(9), 79–85 (1997). <a href="https://doi.org/10.1109/2.612253">https://doi.org/10.1109/2.612253</a>.
- ➤ Jerraya, A.A., Tenhunen, H., Wolf, W.: Guest Editors' Introduction: Multiprocessor Systems-on-Chips. IEEE Computer 38(7), 36–40 (2005). <a href="https://doi.org/10.1109/mc.2005.231">https://doi.org/10.1109/mc.2005.231</a>.
- ➢ Gorder, P.F.: Multicore Processors for Science and Engineering. Computing in Science and Engineering 9(2), 3–7 (2007). <a href="https://doi.org/10.1109/mcse.2007.35">https://doi.org/10.1109/mcse.2007.35</a>.
- Academy, S.: CS201: Words in Computer Architecture Saylor Academy. <a href="https://learn.saylor.org/mod/page/view.php?id=18960">https://learn.saylor.org/mod/page/view.php?id=18960</a>.
- ➢ Gorder, P.F.: Multicore Processors for Science and Engineering. Computing in Science and Engineering 9(2), 3–7 (2007). <a href="https://doi.org/10.1109/mcse.2007.35">https://doi.org/10.1109/mcse.2007.35</a>.

### Bibliografía



- > Quinn, M.A.: Parallel Programming in C with MPI and OpenMP, (2003). <a href="http://www.inf.puc-rio.br/noemi/cd-06/cd3.pdf">http://www.inf.puc-rio.br/noemi/cd-06/cd3.pdf</a>.
- Performance Comparison of MPI Implementations over InfiniBand, Myrinet and Quadrics (2003). https://ieeexplore.ieee.org/document/1592961.
- Chandrashekar, B.N., Shastry, K.A., Manjunath, B.A., Geetha, V.: Performance Model of HPC Application On CPU-GPU Platform\*. (2022). <a href="https://doi.org/10.1109/mysurucon55714.2022.9972737">https://doi.org/10.1109/mysurucon55714.2022.9972737</a>.
- Syberfeldt, A.: A Comparative Evaluation of the GPU vs The CPU for Parallelization of Evolutionary Algorithms Through Multiple Independent Runs (2021). <a href="https://ssrn.com/abstract=3937048">https://ssrn.com/abstract=3937048</a>.
- Pacheco, P., Malensek, M.: An Introduction to Parallel Programming. Morgan Kaufmann, (2021).
- Fast Multidimensional Matrix Multiplication on CPU from Scratch (2022). <a href="https://siboehm.com/articles/22/Fast-MMM-on-CPU">https://siboehm.com/articles/22/Fast-MMM-on-CPU</a>.



## ¡Gracias por su atención!