

# PRACTIA 4 ARQO

grupo:1311

pareja:01

Alexandru, Costinel Glont

Mario Valdemaro, Garcia Roque

### Ejercicio 1:

Copia el espacio de memoria pero no le pasa valor, es decir si tenemos una variable a inicializada a 3 este valor no se va a mantener en la ejecución del hilo, en su lugar puede meter cualquier cosa.

El valor no que se tiene no es el que tenemos declarado, puede ser cualquier valor

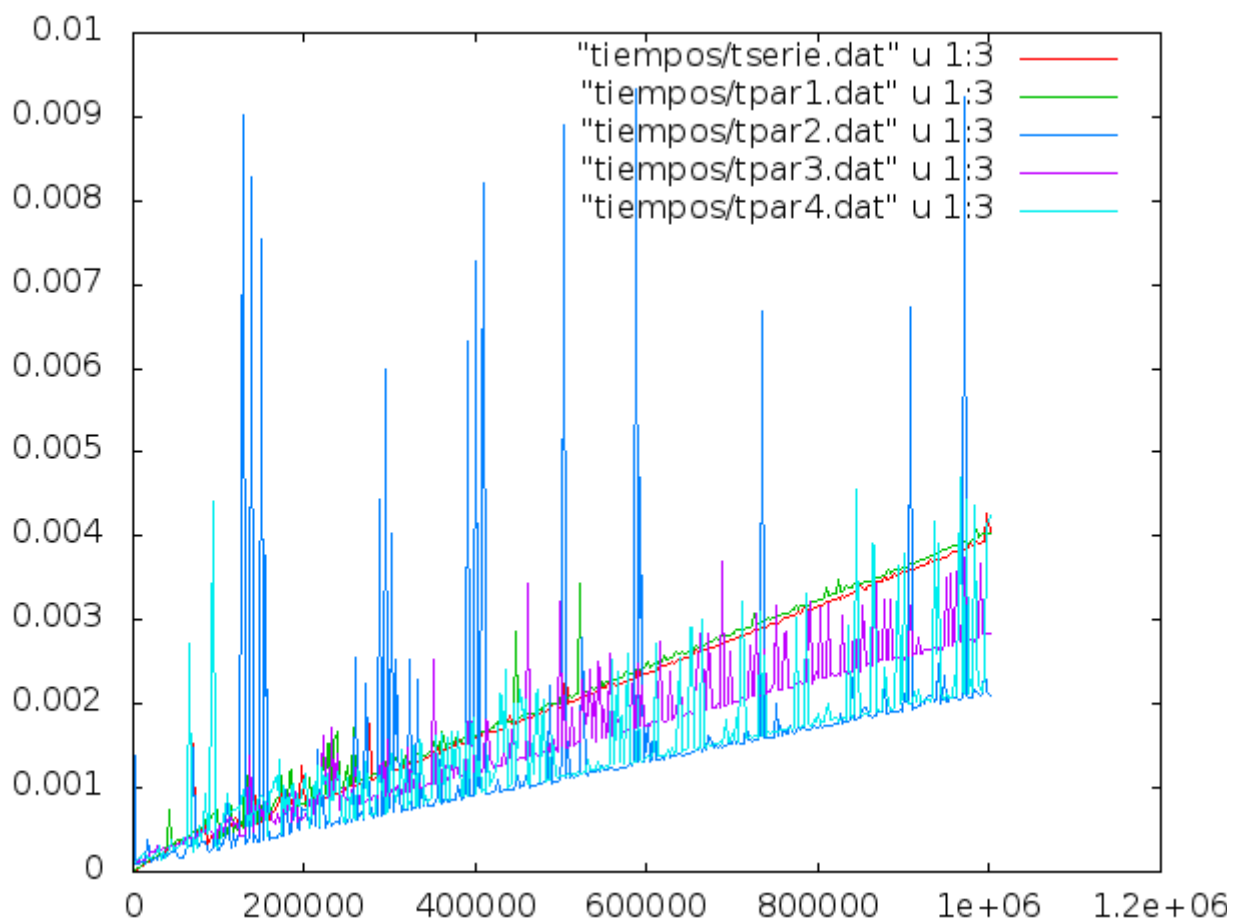
El valor de una función privada no se guarda una vez terminada la ejecución.

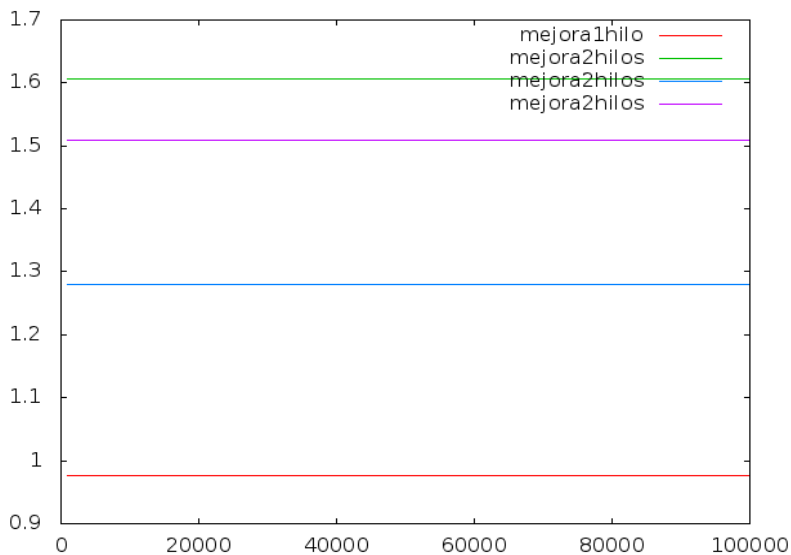
No.

### Ejercicio 2:

1.El resultado no es correcto para `pescala_par1`

2.Se debe al `reduction (+:sum)` esto lo que hace es que una vez ha terminado de ejecutarse la región paralela se va sumando el valor de `sum` de todas las regiones paralelas en una.





Como los tiempos que nos salían para los tamaños de vectores y las repeticiones que nos pedíais no nos parecían una muestra variada, ya que como se observa en la gráfica de tiempos los errores son probables y cambian completamente las gráficas, hemos decidido volver a hacer las gráficas desde valores de 1000 a un millón con incrementos de 2500 y hemos obtenido los datos que se observan.

La segunda gráfica corresponde a las aceleraciones globales con respecto a no usar hilos, donde las líneas azul y morada

son de las mejoras usando 3 y 4 hilos.

Observamos que la mejora mas significativa se produce usando 2 hilos.

¿Por qué los tiempos de 2 hilos son mejores en comparación con 3 o 4 hilos?

Esto se debe a que el ordenador que usamos para tomar las muestras tiene 2 núcleos y puede generar 2 hilos reales. El resto de hilos que se generan son producidos por el software por tanto no son hilos reales producidos por el hardware, así que no nos ahorran tiempo.

¿Por qué los tiempos usando un hilo son peores que no usar hilos?

Como se ve en las gráficas de tiempos usar un hilo y no usarlo es casi equivalente salvo porque al usar un hilo hay un pequeño incremento constante siempre, es decir los tiempos de ejecución de ambos son paralelos. Esto se debe a que el hilo necesita algo mas de tiempo para empezar a ejecutarse, como vemos este tiempo siempre es constante.

### Ejercicio 3:

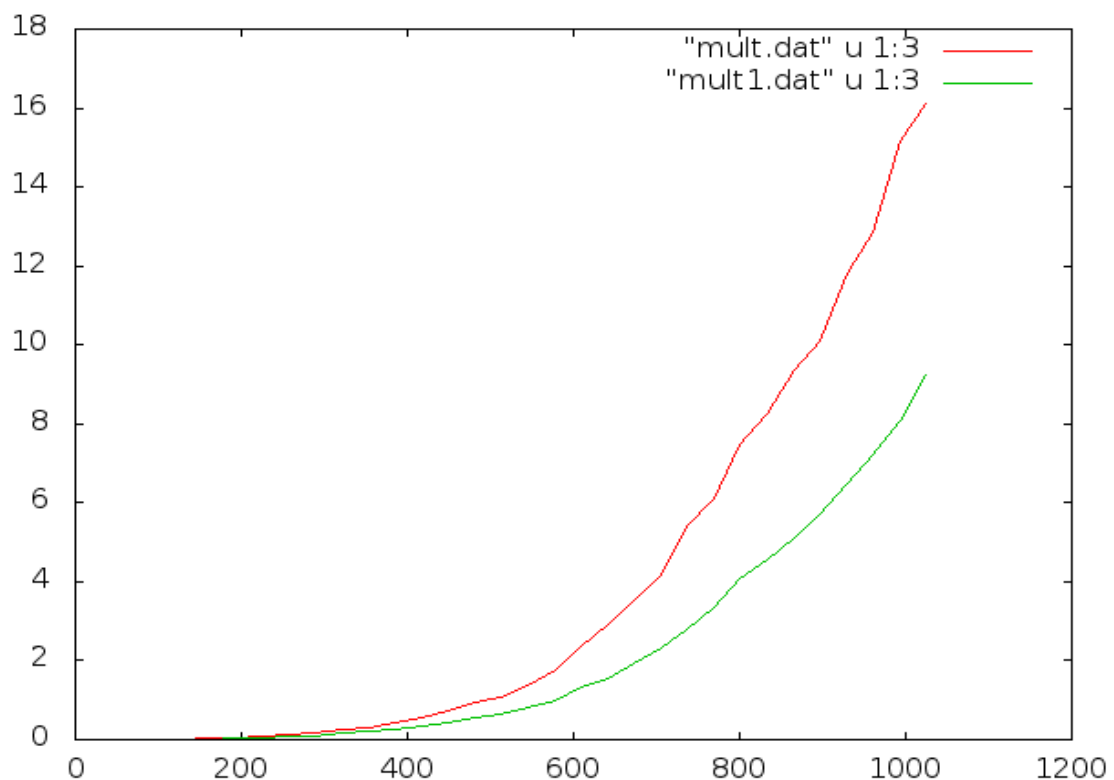
#### TIEMPOS

Hilos	1	2	3	4
Serie	11.13			
Paralela_B1	12.03	6.98	16.59	14.34
Paralela_B2	11.87	6.19	8.29	6.33
Paralela_B3	11.76	5.86	6.05	6.00

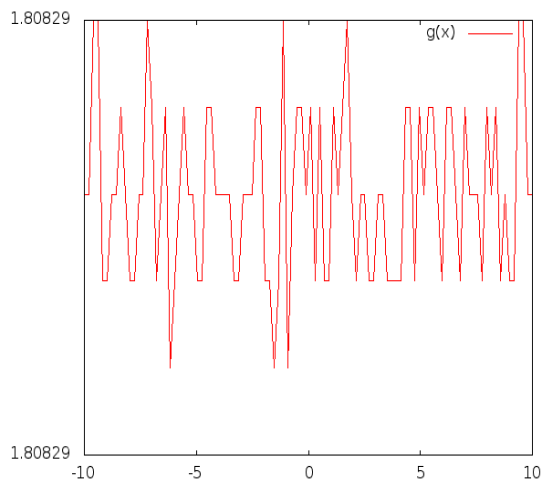
#### SPEEDUP

Hilos	1	2	3	4
Serie	1			
Paralela_B1	0.96	1.59	0.67	0.78
Paralela_B2	0.94	1.80	1.34	1.76
Paralela_B3	0.95	1.90	1.84	1.86

El mejor es el bucle mas externo ya que es el que mas operaciones tiene que realizar por tanto si usamos hilos en este bucle se nos repartirá mejor el trabajo. El peor es el bucle mas interno por el mismo motivo.



Como se ve hay una mejora en los tiempos que se hace mas evidente cuando el tamaño de la matriz es mayor. En los 2 puntos de la tabla se ve como la mejora es casi del doble, esto se debe a que tenemos casi el doble de capacidad de procesamiento ya que tenemos 2 hilos y por tanto aprovechamos todo el hardware de la cpu.



Como vemos la mejora se encuentra al rededor del 1.8 por los que es bastante recomendable usar hilos a la hora de llevar a cabo tareas como estas. Como he dicho antes la mejora es casi del doble. Entendemos que se alrededor de un 20% debido a temas de crear hilos y/o que los hilos no sean tan eficientes como lanzar un proceso.