

Jaime López-Hidalgo (A01638037)
César Alexis Sahagún Navarro (A01637885)
Jesús Palomino Hurtado (A01638492)

“Actividad 5.2 Programación paralela y concurrente”

Implementación de métodos computacionales

20 de mayo de 2021

Problema de programación paralela

La suma de los números primos menores a 10 es:

$$2 + 3 + 5 + 7 = 17$$

Utilizando el lenguaje de programación indicado por tu profesor (Scheme, Racket, Clojure), escribe dos versiones de un programa que calcule la suma de todos los números primos menores a 5,000,000 (cinco millones):

- La primera versión debe ser una implementación convencional que realice el cómputo de manera secuencial.
- La segunda versión debe realizar el cómputo de manera paralela a través de los mecanismos provistos por el lenguaje siendo utilizado (por ejemplo, places o la función pmap). Debes procurar paralelizar el código aprovechando todos los núcleos disponibles en tu sistema.

Ambas versiones del programa deben dar 838,596,693,108 como resultado.

Con el fin de que el proceso de cómputo sea más intenso para el CPU, utiliza el siguiente algoritmo:

Algoritmo para determinar si no es un número primo. Devuelve verdadero o falso.

1. Si no es menor que 2, el algoritmo termina devolviendo falso.
2. Para i desde 2 hasta $\lceil \sqrt{n} \rceil$, realiza lo siguiente:
 - El algoritmo termina devolviendo falso si no es divisible entre i de manera exacta, de otra se repite el ciclo con el siguiente valor de i .
3. El algoritmo termina devolviendo lo verdadero si el ciclo del punto anterior concluyó de manera normal.

Mide el tiempo en que tarda en ejecutar cada versión del programa y calcula el speedup obtenido usando la siguiente fórmula:

$$S_p = \frac{T_1}{T_p}$$

En donde:

- p es el número de procesadores (núcleos).
- T_1 es el tiempo que tarda en ejecutarse la versión secuencial del programa.
- T_p es el tiempo que tarda en ejecutarse la versión paralela del programa utilizando p procesadores. • S_p es el speedup obtenido usando p procesadores.

Escribe un breve documento en donde reportes los resultados obtenidos y entrégalo junto con el código fuente de tus implementaciones.

Resultado

1 thread

```
run:
Hilos: 1

Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 6275 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 6576 ms

Speedup: 1.0479681274900399
```

2 threads

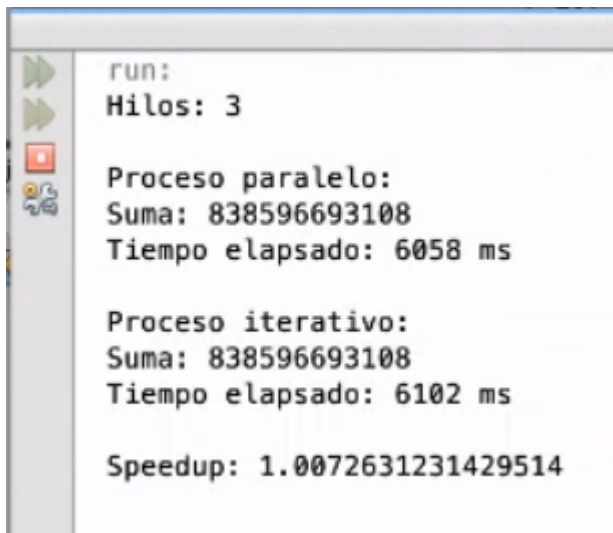
```
run:
Hilos: 2

Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 5920 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 5955 ms

Speedup: 1.005912162162162
```

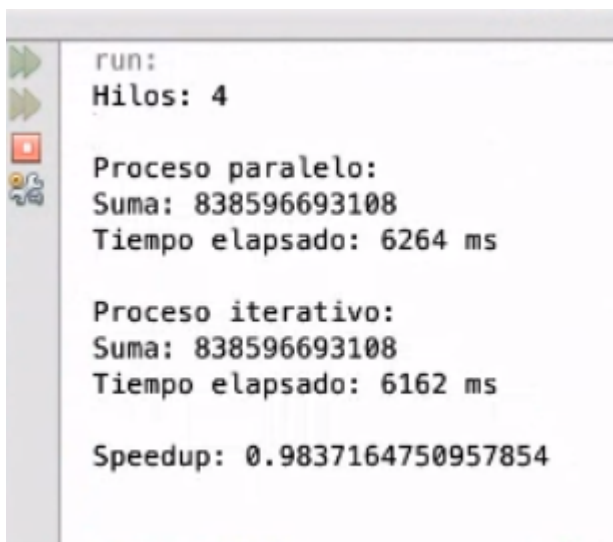
3 threads



The screenshot shows a Java Swing window with a title bar and a standard Mac OS X-style title bar (red, yellow, green buttons). The window contains a text area with the following text:

```
run:  
Hilos: 3  
  
Proceso paralelo:  
Suma: 838596693108  
Tiempo elapsado: 6058 ms  
  
Proceso iterativo:  
Suma: 838596693108  
Tiempo elapsado: 6102 ms  
  
Speedup: 1.0072631231429514
```

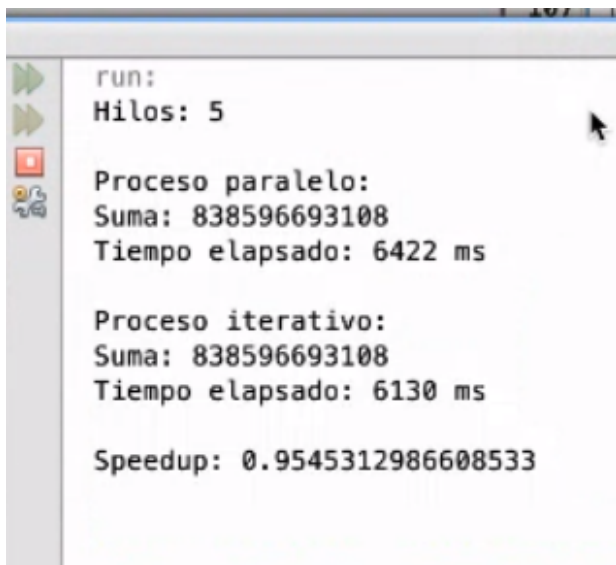
4 threads



The screenshot shows a Java Swing window with a title bar and a standard Mac OS X-style title bar (red, yellow, green buttons). The window contains a text area with the following text:

```
run:  
Hilos: 4  
  
Proceso paralelo:  
Suma: 838596693108  
Tiempo elapsado: 6264 ms  
  
Proceso iterativo:  
Suma: 838596693108  
Tiempo elapsado: 6162 ms  
  
Speedup: 0.9837164750957854
```

5 threads



A screenshot of a Java Swing window with a light gray title bar and a white content area. On the left side, there is a vertical toolbar with four icons: a green double arrow, a green single arrow, a red square, and a yellow speech bubble. The text in the window is as follows:

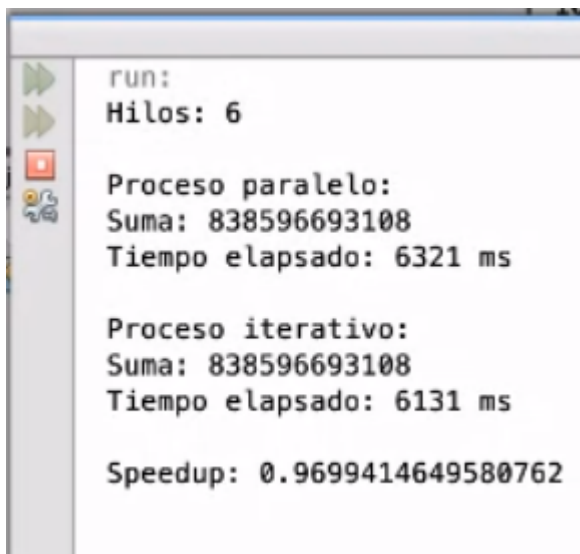
```
run:
Hilos: 5

Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 6422 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 6130 ms

Speedup: 0.9545312986608533
```

6 threads



A screenshot of a Java Swing window with a light gray title bar and a white content area. On the left side, there is a vertical toolbar with four icons: a green double arrow, a green single arrow, a red square, and a yellow speech bubble. The text in the window is as follows:

```
run:
Hilos: 6

Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 6321 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 6131 ms

Speedup: 0.9699414649580762
```

7 threads

```
run:
Hilos: 7
|
Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 6128 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 5970 ms

Speedup: 0.9742167101827677
```

8 threads

```
run:
Hilos: 8
|
Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 5634 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 5583 ms

Speedup: 0.9909478168264111
```

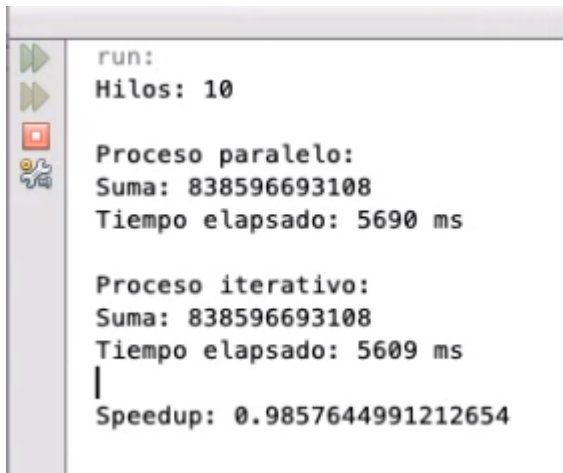
9 threads

```
run:
Hilos: 9
|
Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 5622 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 5643 ms

Speedup: 1.003735325506937
```

10 threads



```
run:
Hilos: 10

Proceso paralelo:
Suma: 838596693108
Tiempo elapsado: 5690 ms

Proceso iterativo:
Suma: 838596693108
Tiempo elapsado: 5609 ms
|
Speedup: 0.9857644991212654
```

Análisis

Como podemos observar, no existe gran diferencia entre los tiempos elapsados para el proceso paralelo e iterativo. Hicimos 10 pruebas, comenzando con 1 hilo, incrementando de 1 en 1. De acuerdo a nuestros resultados, la mayor eficiencia (mayor speedup) se logró al utilizar 3 hilos. La menor eficiencia se logró al utilizar 5 hilos.