

Junio 2022



Tecnológico de Monterrey

Actividad 5.2 Programación paralela y concurrente

**Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Querétaro**

Implementación de métodos computacionales

TC2037.601

Pedro Oscar Pérez Murueta

Presenta:

Said Ortigoza Trujillo | A01707430

Miguel Weiping Tang Feng | A01610836

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 n es un número primo. Devuelve *verdadero* o *falso*.

1. Si n 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 n es divisible entre i de manera exacta, de otra se repite el ciclo con el siguiente valor de i .
3. El algoritmo termina devolviendo *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 (o 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.

Enlace al repositorio de Github con el código:

https://github.com/saidortigoza/Actividad5.2_TC2037

Resultado del speedup al ejecutar las pruebas:

Utilizando programación paralela se logró ejecutar significativamente el tiempo de ejecución, primero se probó un tamaño de 1,000,000, obteniendo el siguiente resultado:

```
Starting...
Resultado en secuencial: 3.755e+010
Avg execution time: 3003
Resultado en paralelo: 3.755e+010
Avg execution time: 2265.7
```

Posteriormente, con un tamaño de 5,000,000, se obtuvo el siguiente resultado:

```
Starting...
Resultado en secuencial: 8.386e+011
Avg execution time: 30029
Resultado en paralelo: 8.386e+011
Avg execution time: 21880
```

Por lo tanto, el speedup utilizando 8 procesadores es de 1.32 para un SIZE de 1 millón y de 1.37 para un SIZE de 5 millones. Por lo que, la programación en paralelo resulta muchísimo más eficiente que la programación secuencial para iteraciones de gran tamaño, ya que las tareas se pueden dividir en bloques y ejecutarse simultáneamente para reducir el tiempo de ejecución.