Modelo de urnas

Francisco Gerardo Meza Fierro

1. Introducción

En esta práctica se intentará paralelizar el código propuesto y se medirá el tiempo que se logra ahorrar, donde la simulación de este código hace referencia a coalescencias y fragmentaciones de procesos, en los cuales sus partículas se unen para formar cúmulos y éstos pueden volver a descomponerse en otros menores.

2. Paralelización del código

Las intenciones que se tienen al querer paralelizar un código son las de minimizar el tiempo en que el código esté corriendo. Al analizar el código propuesto se determinó que solo se paralelizarían las funciones `romper` y `unir`, las cuales se encargan respectivamente de fragmentar y unir los cúmulos. Sin embargo, estas funciones realizan operaciones demasiado sencillas (computacionalmente hablando) y aún así, a comparación del resto de líneas del código, resultaron ser las más viables a paralelizar debido a que son las funciones más significativas del código.

Para ver el ahorro del tiempo que se esperaría tener tras la paralelización, se hizo uso de la función `Sys.time` para medir el tiempo de ejecución del código y se implementó en ambos códigos: el propuesto y el paralelizado. En ambos se varió el tiempo de la simulación (definido en la variable `duracion`). Para cada variación se realizaron veinte réplicas y se promediaron los tiempos.

El siguiente cuadro reúne esos tiempos obtenidos al correr el código propuesto y el código paralelizado.

Cuadro 1: Tiempos promedio en segundos

Duración	Código propuesto	Código paralelizado
10	3.5271	4.0737
40	14.0888	14.8698
160	56.8231	59.5482
640	226.6684	236.6405

La idea ahora es analizar estos tiempos pero ahora variando los valores de n y k, los cuales miden respectivamente el número de partículas y sus tamaños. k se varió en 400, $1,600$, $6,400$, $25,600$ y $102,400$, mientras que n se varió en razón de $30k$. De igual forma, para cada variación se realizaron veinte réplicas y se promediaron los tiempos.

La figura 1 muestra la comparación entre los tiempos obtenidos, donde la línea roja muestra los tiempos del código propuesto y la azul muestra los tiempos del código paralelizado.

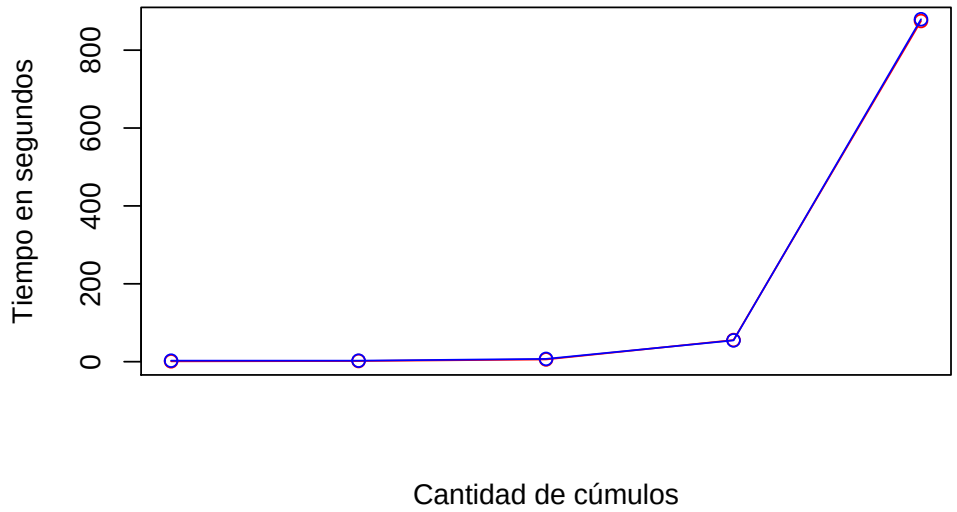

Figura 1: Comparación de tiempos en ambos códigos

3. Conclusiones

La diferencia resultó ser mínima entre ambas implementaciones resultando mejor la secuencial, pese a que en algunas instancias la paralelización resultó mejor; estas diferencias de tiempos oscilaban entre medio segundo y dos minutos.

Esto debido, como ya se mencionó, a la poca complejidad en las operaciones que se efectúan en las funciones que se paralelizaron; debe aclararse que la simulación se llevó a cabo en una computadora usando tres núcleos, es probable que asignarle más núcleos a la misma tarea pueda ayudar a reducir un poco más los tiempos. Aún así, para esta simulación en específico, resulta mejor llevarla a cabo en secuencial.