

Práctica 8

Jose David Martinez Ruiz

3 de octubre de 2017

1. Introducción

En esta práctica número ocho, se implementó el modelo de urnas en la simulación de la creación y destrucción de cúmulos. Lo principal en esta práctica es implementar la paralelización de manera que se ahorre la mayor cantidad de tiempo posible. Se realizó una comparación entre el programa original que se realiza de manera secuencial. De manera que se pueda ver realmente el efecto que hay en el ahorro de tiempo entre una implementación a otra, se corrieron diferentes cantidades de cúmulos al inicio de la simulación y cantidad total de partículas, de manera que la proporción entre cúmulos y partículas se mantenga igual.

Para el primer reto se realizó algo similar a lo anterior, solo que ahora se varió la cantidad de cúmulos al iniciar (k) y se creó una dependencia entre la cantidad total de partículas (n) de manera que treinta veces el número de cúmulos sea el total de partículas en esa simulación. Para cada cantidad inicial de cúmulos se realizaron diez replicas, tanto para la implementación secuencial como la paralela. Se hará un análisis estadístico para verificar si la diferencia de tiempo entre una implementación y otra es significativo.

2. Desarrollo

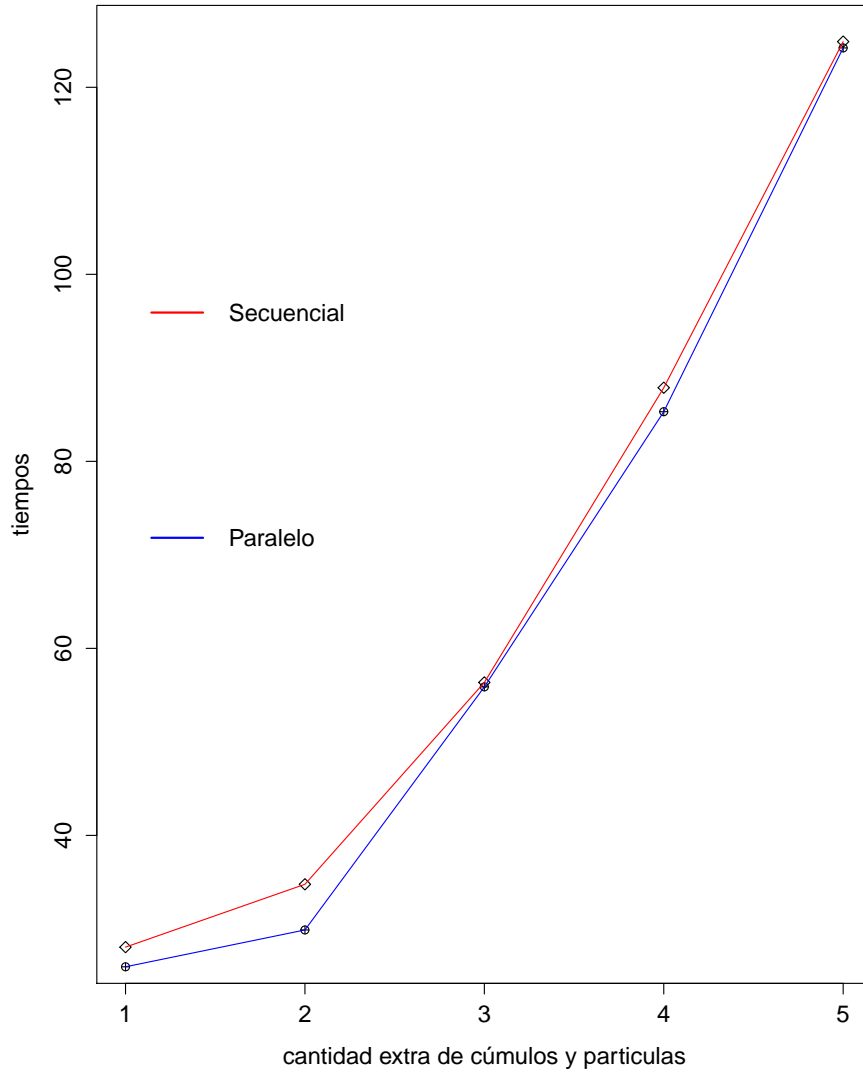
Primero se procedió a verificar el código proporcionado para identificar áreas del código donde se puede implementar la paralelización. Se encontraron que las áreas en donde era viable implementar la paralelización era en donde las partículas se rompían y en donde las partículas se unían. De manera que esas áreas se modificaron para que funcionaran en paralelo.

Al principio se utilizó la librería `doParallel` pero los tiempos que arrojo en correr el programa no fueron favorables, por lo que se decidió en utilizar la librería `parallel`. Se crearon dos funciones a parte de manera que cuando se utilizara la paralelización con la librería `parallel`, ésta funcionara de una mejor manera.

Al realizar las comparaciones de los tiempos se notó que con una duración del experimento tan pequeña como cinco los tiempos en ambas implantaciones no presentaban una diferencia muy notoria, se optó por agrandar la duración de la simulación a treinta pasos, de manera que los efectos de la cantidad de cúmulos y numero de partículas pudieran ser mas notorias.

La cantidad total de partículas es de un millón, y la cantidad de cúmulos es de diez mil, estos valores se multiplicaron por uno hasta por cinco, de manera que dichas multiplicaciones fueron las variaciones para cada caso. Se mantuvo constante la proporción entre la cantidad de cúmulos y el número total de partículas para que así no haya variaciones adicionales en cada simulación.

Los tiempos de ejecución fueron medidos en ambas implementaciones para cada caso. La siguiente gráfica muestra los datos obtenidos en cada caso.



Como se puede observar, para estas cantidades de cúmulos y de número total de partículas la implementación en paralelo resultó ser mejor que la secuencial, aunque las diferencias parezcan ser pequeñas, la implementación en paralelo en cada caso considerado resultó ser la mejor.

3. Reto 1

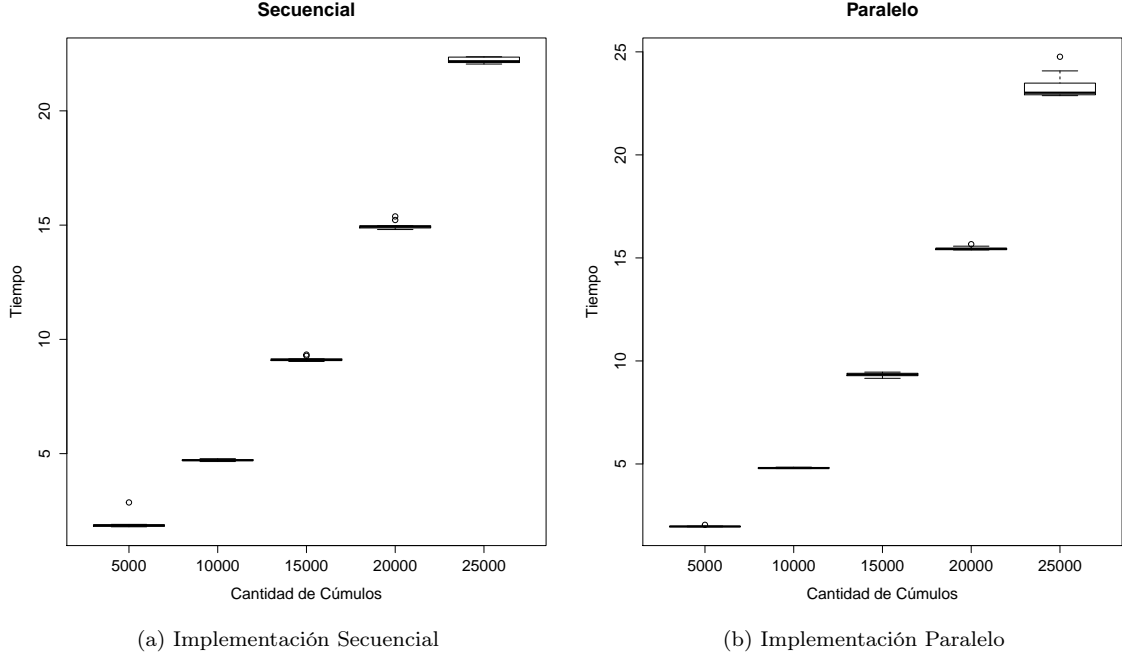
Para este primer reto se busca ver si hay una diferencia significativa entre la implementación en paralelo y la secuencial cuando la cantidad total de partículas es igual a treinta veces la cantidad inicial de cúmulos y la cantidad de cúmulos se encuentra variando.

Para esto se tomó una cantidad de cúmulos inicial de mil y esta cantidad fue multiplicada desde cinco hasta veinticinco con pasos de cinco, de manera que haya una mayor variedad entre las simulaciones. Para cada uno de estos casos se realizaron diez replicas, para de esta manera poder realizar un análisis estadístico. Debido a cómo está definida la cantidad total de partículas la proporción entre la cantidad de cúmulos y el número total de partículas, al igual que en el caso anterior, también se mantiene en cada caso.

Se tomó el tiempo de ejecución de cada implementación para cada replica de cada caso, al final de obtener cada uno de estos datos se procedió a verificar si los tiempos obtenidos se comportaban de una manera normal, para así poder saber que prueba estadística utilizar. Después de realizar

algunas pruebas y de verificar la gráfica de densidad de los tiempos obtenidos, se comprobó que los datos no se comportan de una manera normal, por lo que se utilizaran pruebas no paramétricas.

Los tiempos obtenidos en paralelo, en general son más grandes que los tiempos obtenidos en secuencial, para cada caso que se consideró. A continuación, se muestra una caja de bigotes para implementación.



Como se puede observar, en ambos casos los datos no varían demasiado y los tiempos parecen ser muy similares, aunque los tiempos de la implementación en paralelo son ligeramente mayor que la implementación en secuencial. Para poder ver si hay realmente una diferencia significativa entre ambos casos se procedió a utilizar la prueba de Wilcoxon y se compararon las diez réplicas de cada valor utilizado como cantidad inicial de cúmulos. Las pruebas en general arrojaron que sí hay una diferencia significativa entre las implementaciones, por lo que, en este caso, la implementación secuencial es más rápida.

4. Conclusiones

De la tarea base se concluye que con la proporción que hay entre la cantidad de cúmulos al inicio y número total de partículas, y con esos valores que se le dieron a la cantidad de partículas y de cúmulos que son grandes en comparación a los que se utilizaron en el primer reto, la implementación en paralelo es mejor que la secuencial, esto puede deberse a que llegando a una cierta complejidad en la operaciones, la paralelización es más efectiva que realizar las operaciones de manera secuencial.

Del primer reto se concluye que ambas implementaciones presentan tiempos muy similares, sin embargo, la implementación secuencial resulta ser mejor para este caso. Caso contrario al de la tarea base, en este reto la cantidad de cúmulos y partículas no fueron tan grandes como en la tarea base, por lo que no hay mucho que “procesar”, por así decirlo, por lo que no resulta beneficioso paralelizar en estos casos.

De los dos casos que se estudiaron se concluye que la paralelización puede ahorrarnos tiempo si las tareas a realizar son “pesadas” y que, si las tareas son fáciles de realizar y no presentan un esfuerzo computacional muy grande, la implementación secuencial resulta mejor.