

# Búsqueda Local

Abraham Azael Morales Juárez 1422745

25 de marzo de 2019

## 1. Introducción

Esta práctica se trata sobre fenómenos de coalescencia y fragmentación [2], la coalescencia es la unión de dos elementos conformando uno sólo. Lo que ocasiona que la partícula con el paso del tiempo aumente de tamaño gradualmente [1]. Esto es relevante en distintos ámbitos, como, por ejemplo, en el tratamiento de aguas residuales con el uso de filtros.

## 2. Objetivos

Optimizar el código para que se ejecute en el menor tiempo por medio de paralelización.

Realizar una gráfica para comparar los tiempos obtenidos de ambos métodos y observar si el ahorro es significativo para diferentes combinaciones de  $k$  y  $n$ .

## 3. Resultados

Para la realización de esta práctica se tomaron en cuenta distintos parámetros:

Réplicas de cincuenta.

Número de cúmulos  $k$  de 5,000, 10,000 y 15,000.

Número de partículas  $n$  de 500,000, 1,000,000 y 1,500,000.

La paralelización se realizó en la unión de los cúmulos y en la fragmentación de estos. Se coloca sólo un fragmento del código como ejemplo.

```
1 library(testit)
2 library(parallel)
3 paralelo <- data.frame()
4 k <- 15000
5 n <- 1500000
6 cluster <- makeCluster(detectCores() - 1)
7
8 romperse <- function(tam, cuantos) {
9   romper <- round(rotura(tam) * cuantos)
10  resultado <- rep(tam, cuantos - romper)
11  if (romper > 0) {
12    for (cumulo in 1:romper) {
13      t <- 1
14      if (tam > 2) {
15        t <- sample(1:(tam-1), 1)
16      }
17      resultado <- c(resultado, t, tam - t)
18    }
19  }
20  assert(sum(resultado) == tam * cuantos)
21  return(resultado)
22 }
```

Los resultados obtenidos son distintos a los esperados, debido a como se observa en la Figura 1, si hay diferencia en los tiempos de ambos métodos pero en la primera iteración, que corresponde a  $k$  5,000 y  $n$  500,000, se observa que el método de paralelización tarda más en comenzar y debido a que son pocos datos los que se están analizando, el método secuencial resulta ser el adecuado. Por otra parte, en la segunda y tercera iteración, se observa una reducción en los tiempos de la paralelización como se esperaba, esto es debido al aumento en los datos analizados.

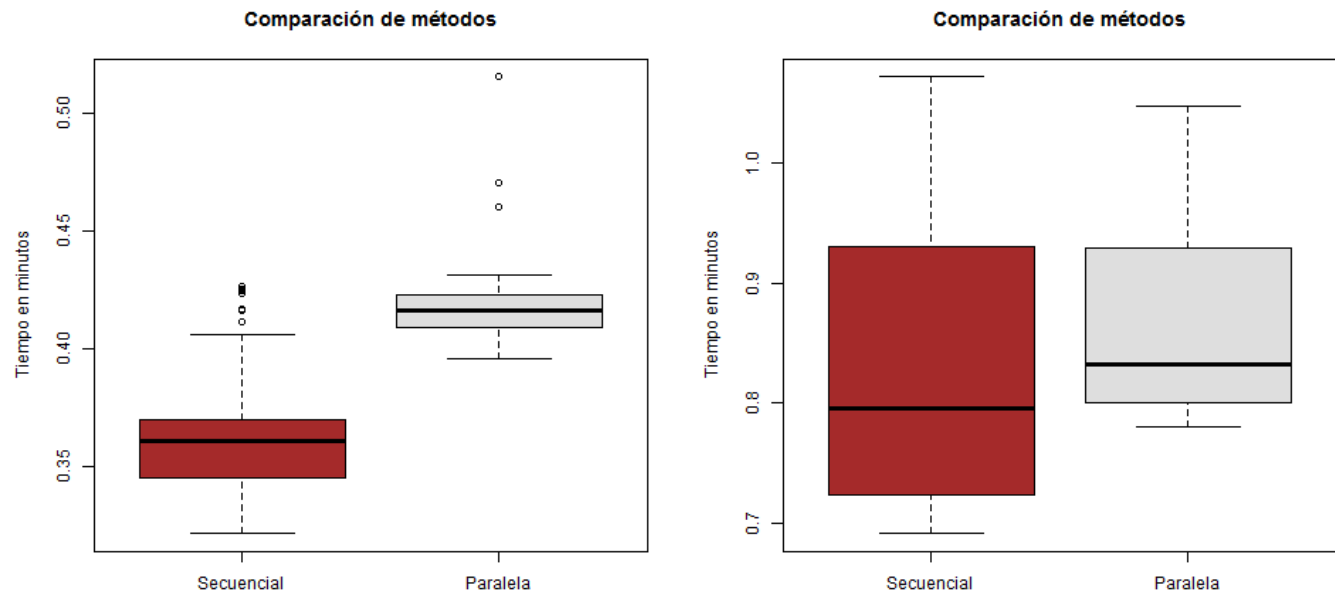


Figura 1: Comparación de tiempos de las iteraciones  $k$  5,000 y de  $n$  500,000 (lado izquierdo) y de  $k$  10,000 y  $n$  de 1,000,000 (lado derecho).

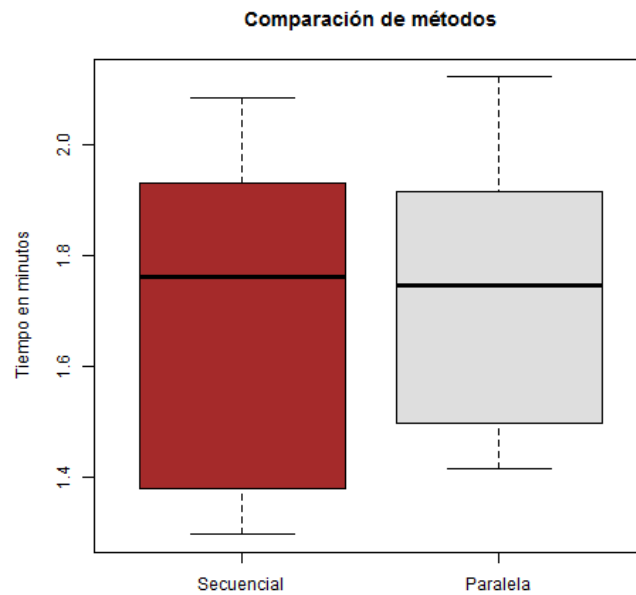


Figura 2: Comparación de tiempos de la iteración  $k$  15,000 y de  $n$  1,500,000.

Además se agregan fragmentos del código que corresponde al principio y al final del análisis.

```
1 for (replicas in 1:50) {
2   inicial1 <- Sys.time()
3   originales <- rnorm(k)
4   cumulos <- originales - min(originales) + 1
5   cumulos <- round(n * cumulos / sum(cumulos))
6   assert(min(cumulos) > 0)
7   diferencia <- n - sum(cumulos)
```

```
1 termino1 <- Sys.time()
2   paralelo <- rbind(paralelo, c(termino1-inicial1, k, n, replicas))
3 }
4 names(paralelo) <- c("Tiempo", "Moleculas", "Cumulos", "Replicas")
5 paralelo$Nivel <- "P"
6 totaltiempos <- rbind(secuencial, paralelo)
7 png("Comparacion de metodos.png")
8 boxplot(totaltiempos$Tiempo~totaltiempos$Nivel, main = "Comparacion de metodos", ylab = "Tiempo en
   minutos", col = c("brown", "grey87"), names = c("Secuencial", "Paralela"))
9 graphics.off()
```

## 4. Conclusiones

El método de paralelización es efectivo y rápido en comparación del secuencial, sólo si la cantidad de datos es muy grande, lo contrario ocurre cuando los datos son pocos.

Se logró la paralelización del código y se demostró la importancia de realizar este método.

El método tarda en comenzar debido a que primeramente se dividen las tareas en todos los núcleos para posteriormente realizar el análisis.

## Referencias

- [1] Alexia Ibarra. Práctica 8: Modelo de urnas. *Universidad Autónoma de Nuevo León*, 2018.
- [2] Satu Elisa Schaeffer. Práctica 8: Modelo de urnas, 2019. URL <https://elisa.dyndns-web.com/teaching/comp/par/p8.html>.