

# Reporte práctica ocho: Modelo de urnas

José Anastacio Hernández Saldaña

Posgrado de Ingeniería de Sistemas

1186622

jose.hernandezsld@uanl.edu.mx

3 de octubre de 2017

## Resumen

Este es un reporte sobre la práctica ocho con respecto al tema de modelo de urnas que se realizó en la clase de Simulación de sistemas, cómputo paralelo en R.

## 1. Tarea y Extra Uno: Paralelización de un modelo de urnas

Un modelo de urnas es aquel que trata de simular un contenedor con elementos, para calcular la probabilidad de extraer un elemento de la urna dada alguna característica del elemento; por ejemplo, podemos tener una urna con  $b_b$  bolas color blanco y  $b_a$  bolas color azul de donde se desea conocer la probabilidad de sacar una bola blanca.

Para esta práctica se estudió un modelo donde se tienen  $n$  partículas con  $k$  diferentes tamaños existentes los cuales siguen una distribución normal, existen dos procesos que modifican sus tamaños, uno de división y otro de unión, cada partícula tiene una probabilidad  $p_u$  de unirse a otra y una probabilidad  $p_d$  de dividirse en 2 partes, estas probabilidades están en función de su tamaño a partir de las fórmulas  $p_u(x) = e^{-\frac{x}{c}}$  y  $p_d(x) = \frac{1}{1+e^{\frac{c-x}{d}}}$  donde  $c$  es la media y  $d$  la desviación estándar de los tamaños de las partículas en la distribución. Estos procesos se realizan en cada iteración de la simulación, y se tiene una duración fija.

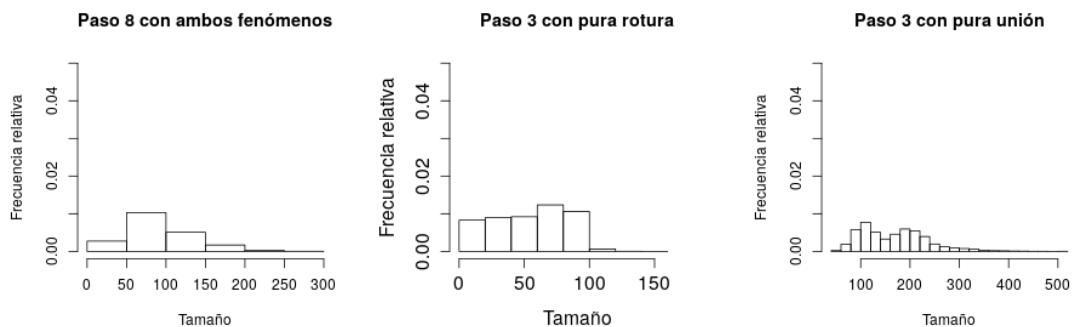


Figura 1: Ejemplos de distribución de tamaños por proceso

Para el experimento se tomó como base el código de la página del curso. Esta implementación del código es iterativa, el primer objetivo es paralelizar lo más que se pueda este código y para la actividad extra ver la comparación en tiempo de cada implementación para saber cual fue mejor.

### 1.1. Diseño del Experimento

Para la paralelización del código, se tenían tres posibles partes del código para aplicar la paralelización.

1. Rutina de cálculo y división de partículas.
2. Rutina del cálculo de cantidad de partículas a unir.
3. Rutina de unión de partículas.

Estas tres rutinas se realizan para cada partícula de un determinado conjunto y sin depender de alguna otra, por lo que las hace adecuadas para paralelizar.

En la rutina 1, se optimizó la división de partículas, haciendo los cálculos por medio de vectores, para la rutina 2 se dejó igual y para la rutina 3 se paralelizó el proceso de unión ya que se identificó que era el más tardado.

Para comenzar la experimentación, se utilizó una computadora con las siguientes especificaciones, Procesador Intel Core i7-4790 CPU @ 3.6GHz  $\times$  8 y memoria RAM de 24 GB utilizando solamente los cuatro núcleos físicos disponibles. Tomando en cuenta los cambios anteriores, se diseñó el experimento con los siguientes parámetros, los valores de  $k$  utilizados fueron: 1 000, 5 000, 10 000, 15 000, 20 000, 25 000, 50 000, y 100 000, los valores de  $n$  eran proporcionales a  $k$  tal que  $n = 30k$ , se utilizaron 30 replicas para cada experimento con simulaciones de duración de 25 iteraciones. El código está disponible en el repositorio git del curso bajo el nombre de Tarea8.r.

## 1.2. Resultados

Los resultados de la experimentación los podemos ver en la figura 2 y en la figura 3, donde el eje vertical nos indica el tiempo de ejecución en segundos y el eje horizontal el tamaño  $k$ . Los resultados se dividieron en 2 graficas para no perder detalle en la escala ya que para valores grandes de  $k$  partículas, el tiempo entre la implementación secuencial y la paralela es muy grande.

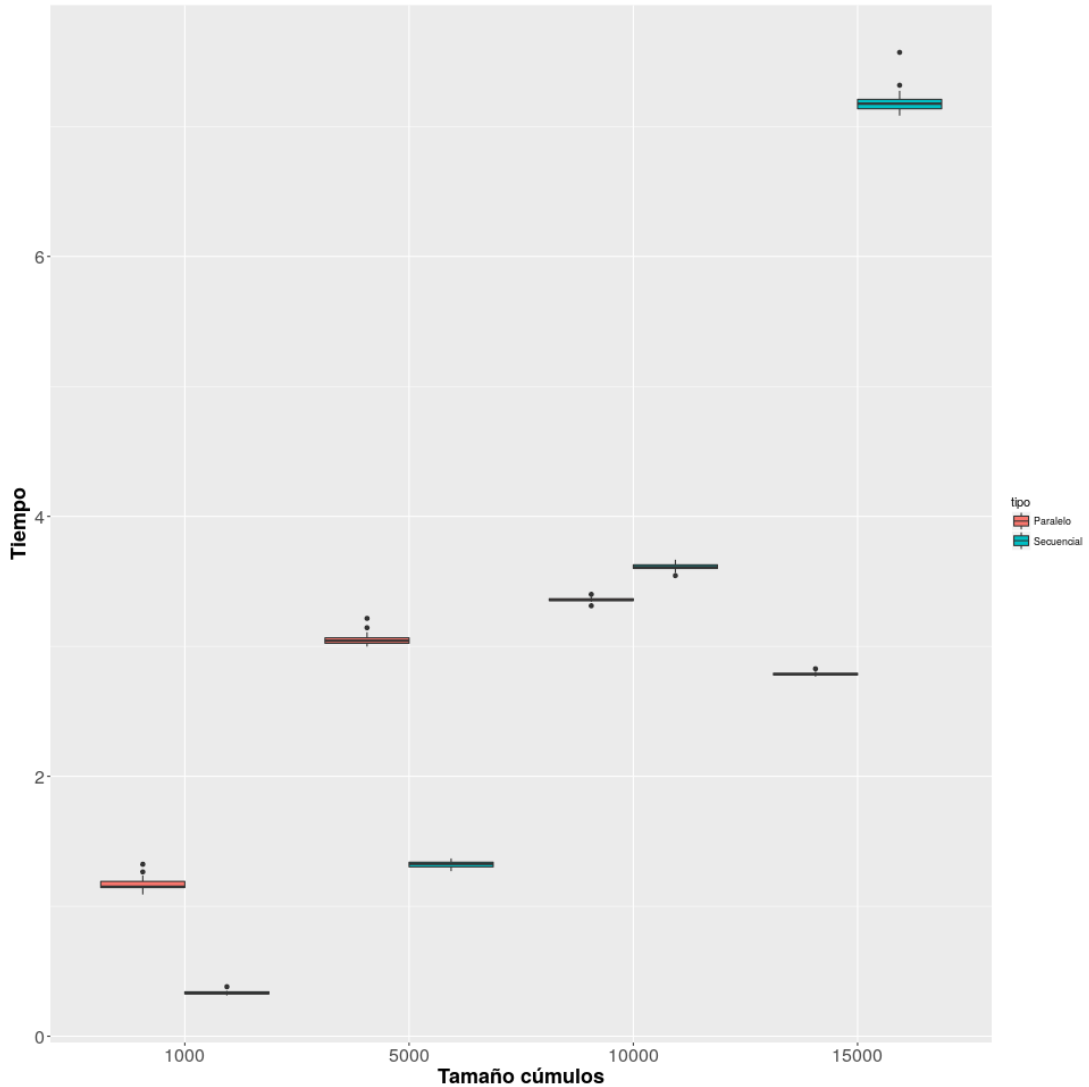


Figura 2: Primeros resultados donde tamaño  $\leq 15000$

En estos resultados podemos ver que para valores de  $k$  mayores a 10 000 tenemos una mejora significativa para la implementación paralela, mientras que para valores menores a 10 000, tenemos que la implementación secuencial es más rápida.

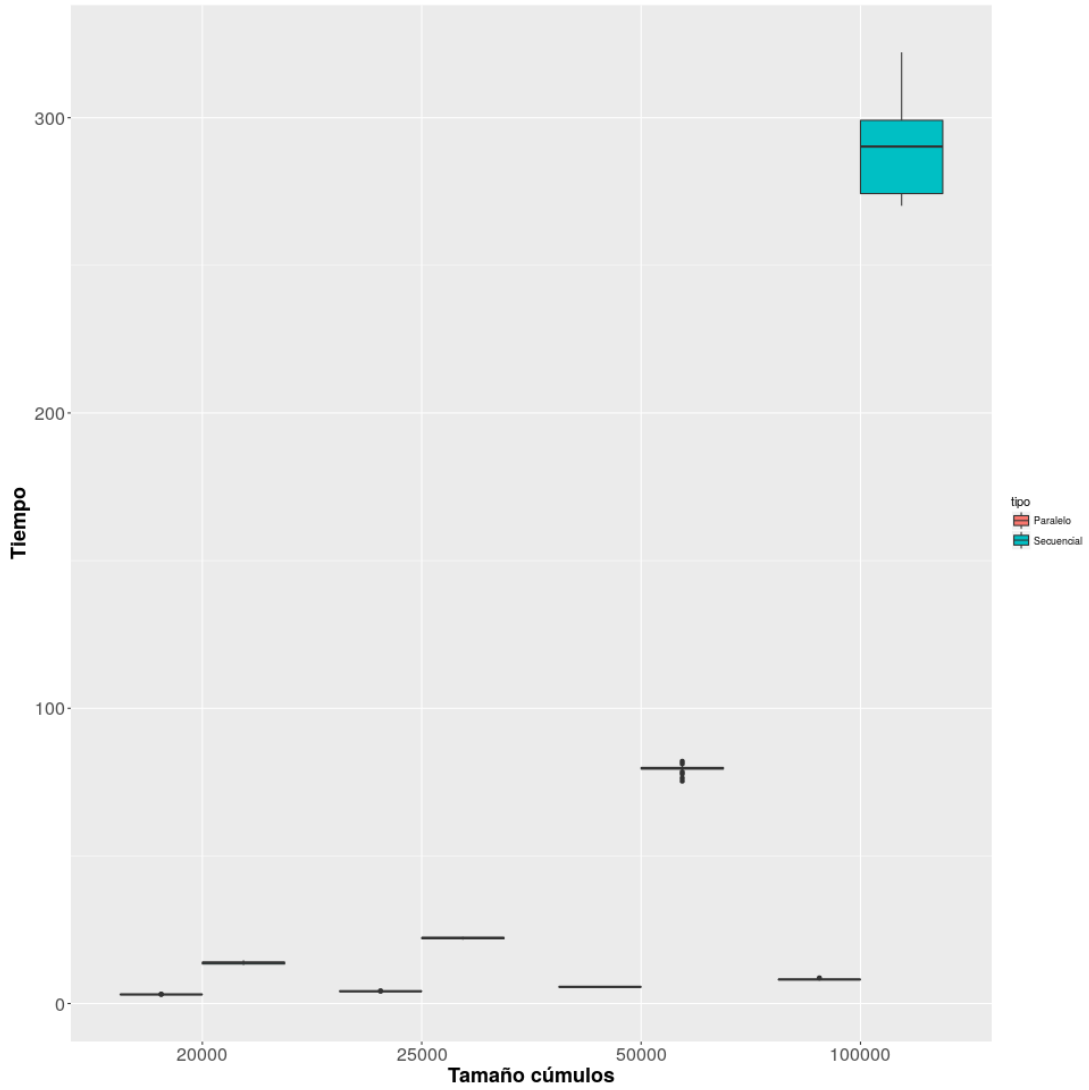


Figura 3: Segundos resultados donde tamaño > 15000

Para poder comparar mejor cada uno de los resultados en cada uno de los niveles, se transformaron los datos, donde cada nivel era dividido por el máximo valor encontrado y tener una escala del cero al uno, donde el tiempo uno es el mayor tiempo que tomo una rutina para completar la simulación. Estos resultados los podemos ver en la figura 4 y aqui podemos apreciar de una manera clara como es la diferencia entre los 2 procedimientos ya que para los últimos valores de  $k$  tenemos mejoras con tiempos 4 veces menores que el tiempo del algoritmo secuencial.

Para corroborar el resultado obtenido se realizó una prueba Wilcoxon donde probaremos la hipótesis nula de la igualdad en la medias de los dos procedimientos, donde se obtuvo un valor  $p$  de  $1.462 \times 10^{-14}$  por lo que se rechaza la hipótesis nula de que las medias son iguales.

## 2. Conclusiones

Se pudo observar las diferencias entre los procedimientos secuencial y paralelo, donde se puede apreciar como para valores pequeños de trabajo, la rutina secuencial es más rápida que la opción paralela, y a partir de un valor ambas muestran una transición de fase, de tal forma que la opción paralela es mas eficiente que la opción secuencial.

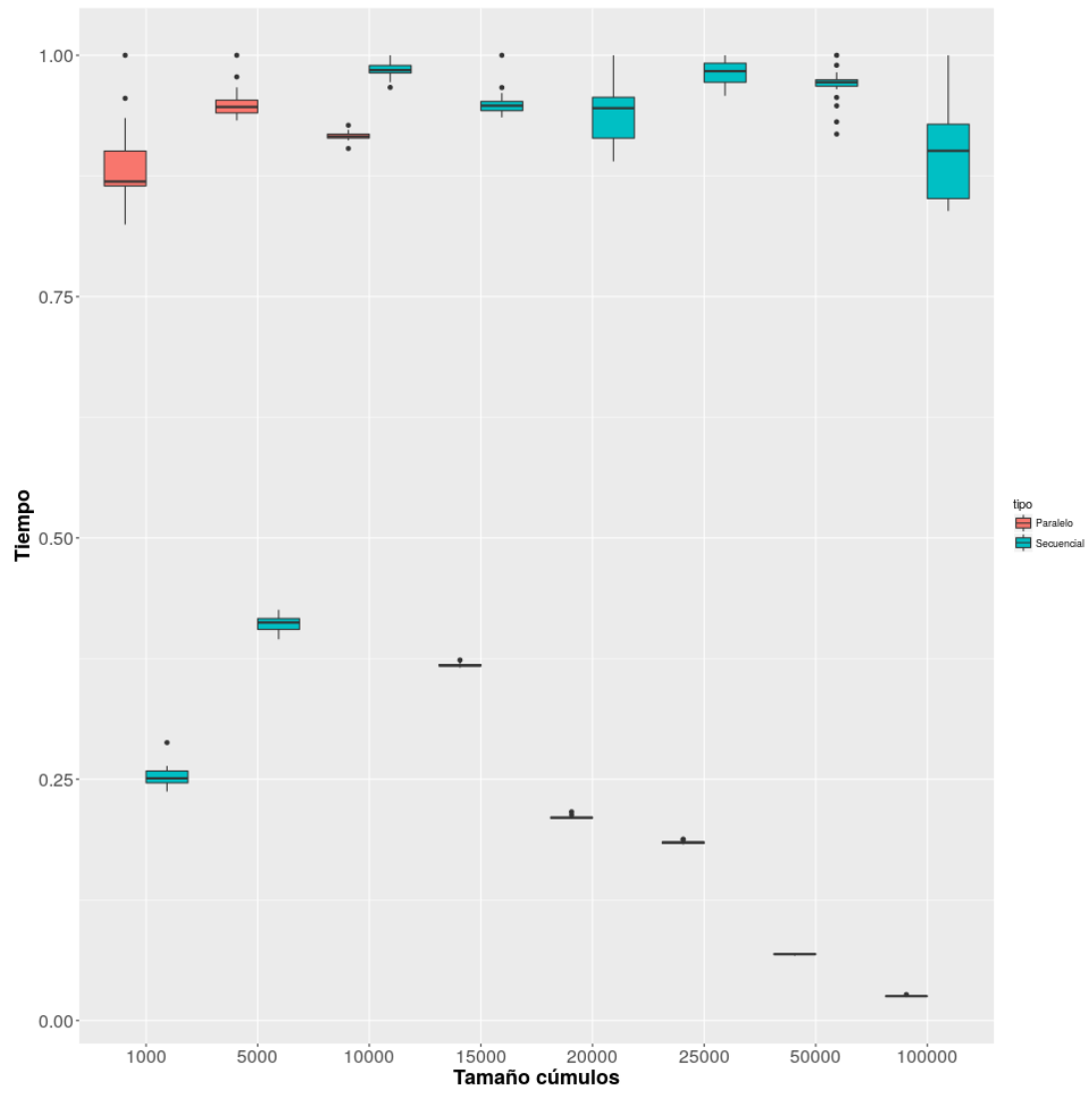


Figura 4: Resultados normalizados