

# Simulación de Sistemas

## Práctica 8

Modelo de urnas  
1455175: Ángel Moreno

1 de octubre de 2018

### Resumen

La octava práctica es sobre fenómenos de coalescencia y fragmentación, donde partículas se unen para formar cúmulos y estos se pueden volver a descomponer en fragmentos más pequeños. La población de cúmulos esta en una tabla de frecuencias para saber cuántos hay de cada tamaño, ésto es típico en los llamados modelos de urnas. La simulación que se implementa avanza por dos fases: fragmentación y coalescencia, se tiene el código de la simulación en forma secuencial y parte de la práctica es implementar una simulación en paralelo para estudiar el ahorro en el tiempo computacional. Se darán a conocer resultados relevantes que se pueden tomar en consideración para futuras simulaciones en paralelo, como por ejemplo no es eficiente paralelizar cálculos muy pequeños.

## 1. Introducción

Se declara una cantidad total de  $n$  partículas y el tamaño de los  $k$  cúmulos sigue una distribución normal. Se declara un tamaño crítico  $c$  que es igual a la mediana de los tamaños iniciales de los cúmulos para modelar la probabilidad de rotura mediante una curva sigmoideal. Para la probabilidad de unión de los cúmulos se utiliza la distribución exponencial con el fin de que cúmulos muy pequeños se junten con mayor fuerza, pero la fuerza de unión sea muy poca cuando sean de tamaños mayores.

La simulación avanza por dos fases en cada iteración. Primero todos aquellos cúmulos que quieren fragmentarse, luego con los cúmulos existentes que quieran unirse, se apuntan para uniones.

### 1.1. Tarea

La tarea se trata de paralelizar tanto como resulte eficientemente posible en esta simulación y medir cuánto tiempo se logra ahorrar, verificando bajo cuáles condiciones el ahorro logrado es estadísticamente significativo con los valores fijos de  $k$  y  $n$  del código ejemplo.

## 2. Simulación

Se implementa la simulación con  $n = 1000000$  partículas y  $k = 10000$  cúmulos fijos, variando cantidad de iteraciones en un cantidad exponencial en base 2 con exponentes desde 1 hasta 8 con 30 réplicas.

Se utiliza el paquete de paralelizado `foreach` en las funciones:

```
1 faserotura <- function(i) {
2   urna <- freq[i,]
3   if (urna$tam > 1) { # no tiene caso romper si no se puede
4     return(romperse(urna$tam, urna$num))
5   } else {
6     return(rep(1, urna$num))
7   }
8 }
9
10 faseunion <- function(i) {
11   urna <- freq[i,]
12   return(unirse(urna$tam, urna$num))
13 }
14
15 fasejuntarse <- function(i) {
```

```

16  return(juntarse[2*i-1] + juntarse[2*i])
17  }

```

## 2.1. Resultados

La figura 1 muestra los resultados de la simulación.

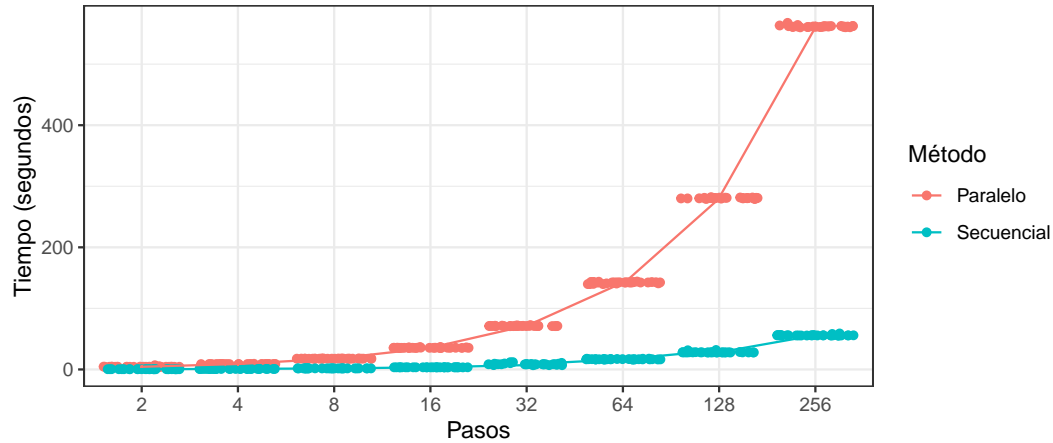


Figura 1: Resultados de los tiempos con 30 réplicas de distintos pasos.

Se observa que el tiempo crece de forma exponencial con el método paralelo mientras que en el método secuencial su comportamiento es lineal. Con los resultados obtenidos se concluye que el paralelo no es tan eficiente como lo es el secuencial, a esto se le puede atribuir a que la cantidad de cálculos son muy pocos y se demore más en la asignación de tareas a los núcleos que resolverlos.

## 3. Reto 1

### Referencias

- [1] SCHAEFFER E. *R paralelo: simulación y análisis de datos*, 2018.  
<https://elisa.dyndns-web.com/teaching/comp/par/>
- [2] VALDES E. *Repository of Github*, 2017.  
<https://github.com/eduardovaldesga/SimulacionSistemas>
- [3] SAUS L. *Repository of Github*, 2018.  
<https://github.com/pejli/simulacion>