

Práctica 8: modelo de urnas

Ricardo Rosas Macías

21 de mayo de 2019

1. Introducción

El modelo de urnas arriba los fenómenos coalescencia y fragmentación, de modo en que une las partículas que se encuentran divididas en dos o más pedazos más pequeños; siempre siendo un entero. Este experimento tiene una reacción parecida a los agentes coagulantes en donde las partículas más pequeñas se agrupan para formar una más grande y así poder ser separada de una manera más fácil.

2. Objetivo

Se realizó cambios en el código proporcionado en la página web [2], de manera que al paralelizar se vuelva más eficiente realizando varias tareas al mismo tiempo y así reducir el tiempo total de la ejecución del código, asimismo se busca visualizar el cambio al generar repeticiones del experimento.

2.1. Descripción

La finalidad del experimento es [2]:

“Paralelizar tanto como resulte eficientemente posible en esta simulación y medir cuánto tiempo se logra ahorrar, de igual manera evaluar si el ahorro es estadísticamente significativo para diferentes combinaciones de partículas y cúmulos.

Para el primer reto, se determinó en cuál momento (en términos del número de iteraciones de la simulación) la fracción de éstas alcance un máximo. Calcula y gráfica esta fracción a lo largo de la simulación. Realiza 30 réplicas para determinar si el número de paso en el cual se alcanza el máximo tiene un comportamiento sistemático.”

3. Resultados y conclusiones

En base al trabajo anteriormente reportado [1], se realizó el código que se muestra en la parte inferior. En las primeras líneas del código se definió los parámetros de experimentación con las cuales se trabajaría, asimismo se ejecutó una paralelización para que el experimento fuera más eficiente. Además se realizó

una comparación de la ejecución del experimento en donde se ponderaron diferentes valores para las partículas que están representadas como “k” y para los cúmulos definidos con la variante “n”, como se muestra en el cuadro 1.

Cuadro 1: Combinación de valores de variables del experimento

Nombre	n	k
C1	100	10000
C2	1000	100000
C3	10000	1000000

```

1 k <- c(100, 1000, 10000)
2 n <- c(10000, 100000, 1000000)
3
4 suppressMessages(library(doParallel))
5 registerDoParallel(clust)
6 cluster <- makeCluster(detectCores()-1)
7 clusterExport(cluster, "romperse")
8 clusterExport(cluster, "rotura")
9 clusterExport(cluster, "funromperse")
10 clusterExport(cluster, "c")
11 clusterExport(cluster, "d")
12 clusterExport(cluster, "assert")
13 clusterExport(cluster, "fununirse")
14 clusterExport(cluster, "union")
15 clusterExport(cluster, "unirse")
16 TimeF <- NULL
17 TimeF <- microbenchmark(exect2 = {
18   originales <- rnorm(k)
19   cumulos <- originales - min(originales) + 1
20   cumulos <- round(n * cumulos / sum(cumulos))
21   assert(min(cumulos) > 0)
22   diferencia <- n - sum(cumulos)
23   if (diferencia > 0) {
24     for (i in 1:diferencia) {
25       p <- sample(1:k, 1)
26       cumulos[p] <- cumulos[p] + 1
27     }
28   } else if (diferencia < 0) {
29
30     for (i in 1:-diferencia) {
31       p <- sample(1:k, 1)
32       if (cumulos[p] > 1) {
33         cumulos[p] <- cumulos[p] - 1
34       }
35     }
36   }
37 }
38 for (paso in 1:duracion) {
39   assert(sum(cumulos) == n)
40   cumulos <- integer()
41
42   for (i in 1:dim(freq)[1]) { # fase de rotura

```

```

43 urna <- freq[i,]
44 if (urna$tam > 1) { # no tiene caso romper si no se puede
45   cumulos <- c(cumulos, romperse(urna$tam, urna$num))
46 } else {
47   cumulos <- c(cumulos, rep(1, urna$num))
48 }
49 }
50 }
51 }, times = 10, unit = "s")
52 stopCluster(clust)
53 print(Ptime)
54 print(TimeF)
55 tiempos <- rbind(Ptime, TimeF)
56 tiempos <- data.frame(tiempos)
57 tiempos$time <- tiempos$time/10**9
58 tiempos$expr <- as.factor(tiempos$expr)

```

Como se menciona en el apartado anterior, para observar cambios significativos en el tiempo del experimento se vario los valores de las partículas y cúmulos respectivamente, de manera que los datos proporcionados se muestran en la gráfica 1, en donde se puede observar claramente que los tiempos al paralelizar son mejores a comparación de la ejecución normal. Por lo tanto es un factor determinante para optar por el uso de dicha función.

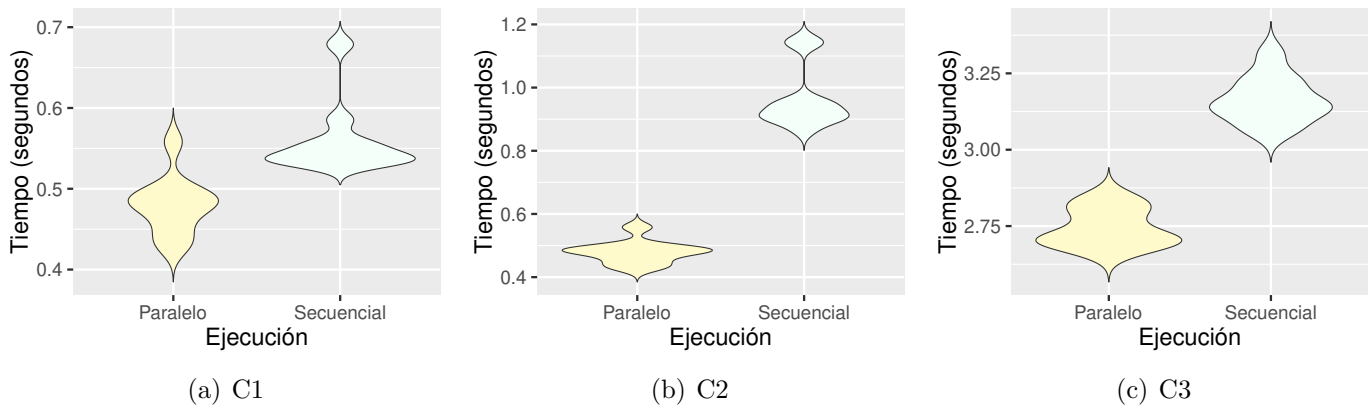


Figura 1: Tiempo del experimento

3.1. Reto 1

Para el primer reto se creó un código en el cual se pueda replicar por 30 veces, como se muestra en las líneas de la parte inferior. Asimismo se usaron las combinaciones anteriores para determinar si el experimento tiene un comportamiento sistemático.

```

1 k <- c(100, 1000, 10000)
2 n <- c(10000, 100000, 1000000)
3
4 funromperse <- function(i){
5   return(as.vector(romperse(as.numeric(freq[i,][1]), as.numeric(freq[i,][2]))))

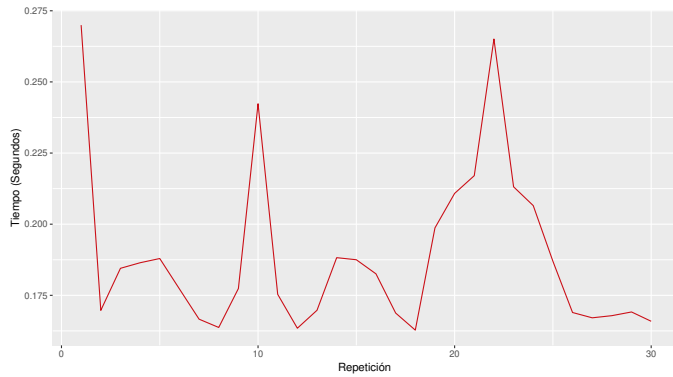
```

```

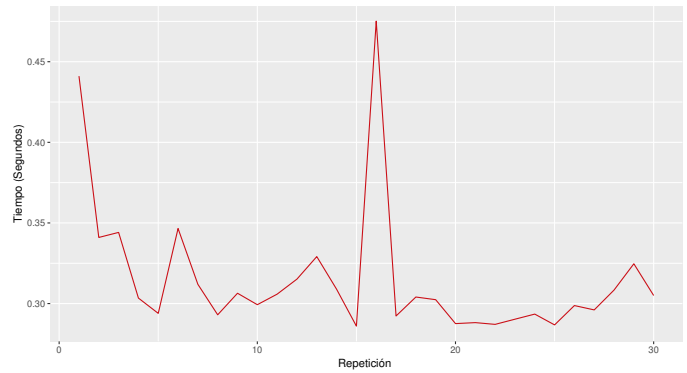
6 }
7 fununirse <- function(i){
8   return( unirse( as.numeric(freq[i,][1]) , as.numeric(freq[i,][2])) )
9 }
10
11 cluster <- makeCluster(detectCores()-1)
12 clusterExport(cluster, "romperse")
13 clusterExport(cluster, "rotura")
14 clusterExport(cluster, "funromperse")
15 clusterExport(cluster, "c")
16 clusterExport(cluster, "d")
17 clusterExport(cluster, "assert")
18 clusterExport(cluster, "fununirse")
19 clusterExport(cluster, "union")
20
21
22 vartime <- data.frame()
23 for (repeticiones in 1:30) {
24   tiempoinicial <- Sys.time()
25   for (paso in 1:duracion) {
26     assert(sum(cumulos) == n)
27     cumulos <- integer()
28     clusterExport(cluster, "freq")
29     cumulos <- as.vector(parSapply(cluster, 1:(dim(freq)[1]), funromperse))
30     a <- c()
31     for(i in 1:length(cumulos)){
32       a <-c(a, cumulos[[i]])
33     }
34     cumulos <- a
35
36     clusterExport(cluster, "freq")
37     cumulos <- as.vector(parSapply(cluster, 1:(dim(freq)[1]), fununirse))
38     a <- c()
39     for(i in 1:length(cumulos)){
40       a <-c(a, cumulos[[i]])
41     }
42     cumulos <- a

```

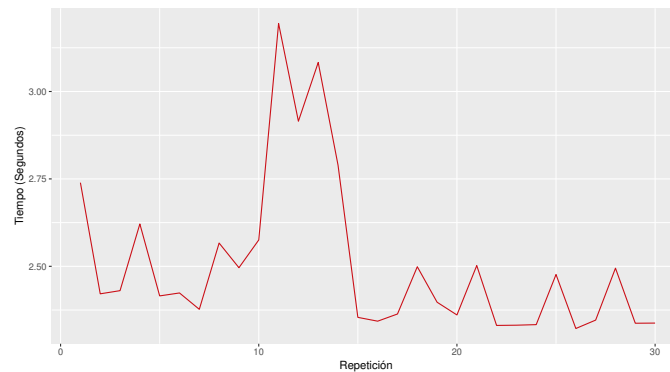
En la figura 2 se muestra la variación del máximo de las repeticiones de la ejecución, las cuales demuestran una tendencia a disminuir la variación en el tiempo de ejecución de las replicas, debido al incremento en los valores de la combinación de partículas y cúmulos. De manera que se puede determinar que el experimento tiene un comportamiento sistemático.



(a) Tiempo de C1



(b) Tiempo de C2



(c) Tiempo de C3

Figura 2: Iteraciones de la simulación

Referencias

- [1] Astrid González. Exercise 8, 2018. URL <https://sourceforge.net/projects/simulacion-de-sistemas/files/Practica%208/>.
- [2] Elisa Schaeffer. Práctica 8: modelo de urnas, 2019. URL <https://elisa.dyndns-web.com/teaching/comp/par/p8.html>.