

Modelo de urnas

Edson Raúl Cepeda Márquez

26 de marzo de 2019

1. Objetivo

El objetivo principal de esta práctica es hacer uso del paralelismo para analizar y comparar los tiempos de ejecución de una simulación de un fenómeno de coalescencia y fragmentación en el que se implementa un modelo de urnas y se varían sus parámetros. Se hace uso del código escrito en el lenguaje de programación R [1] así como también el material de apoyo disponible en la página [2] de la Dra. Elisa Schaffer.

2. Desarrollo

Para fines de comparación se hace una implementación sin paralelismo de las variaciones en el número total de partículas n y el tamaño de los cúmulos existentes k . Primero se encapsula el bloque de código para su uso posterior.

```
1 simulacion <- function(kvar, nvar) { #Encapsular el bloque de codigo
```

A esta función se le pasan los parametros que varian n y k . Para eso se realizan dos ciclos en los cuales estas dos cantidades varían en un rango y además se incluye un ciclo extra de réplicas del experimento para su posterior análisis. Los tiempos se toman con la funcion *system.time* y se guarda en un vector de nombre *tiemposSP* que hace referencia a el tiempo que toma el código sin paralelización.

```
2 replicas <- 10 #numero de replicas
3 tiemposSP <- c()
4 nseq <- seq(1000000,2500000,500000) #variacion del numero total de particulas
5 kseq <- seq(10000,25000,5000) #variacion del numero de cumulos existentes
6 for(nvar in nseq) { #ciclo de n
7   for(kvar in kseq) { #ciclo de k
8     for(rep in 1:replicas) { #ciclo de replicas
9       tsp <- data.matrix(system.time(simulacion(kvar,nvar))[1])[1] #Obtener tiempo
10      tiemposSP <- c(tiemposSP, tsp) #se guarda tiempo
11    }
12  }
13 }
```

Ahora se procede a hacer la implementación paralela. Se usa la función encapsulada previamente y se ejecuta de manera paralela. Los ciclos de la variación de el número total de partículas n y el número de cúmulos existentes k permanecen puesto que en esta implementación se requiere comparar con las mismas variaciones. El número de réplicas se especifica dentro de la función que ejecuta en paralelo. Los resultados se guardan en un vector de nombre *tiemposCP* que hace referencia al tiempo que toma el código ejecutando en paralelo.

```

1 #Implementacion paralela
2 suppressMessages(library(doParallel))
3 cls <- makeCluster(detectCores() - 3)
4 registerDoParallel(cls)
5 tiemposCP <- c()
6 for(nvar in nseq) { #ciclo de n
7   for(kvar in kseq) { #ciclo de k
8     tcp <- foreach(r= 1:replicas, .combine = c) %dopar% data.matrix(system.time(simulacion(kvar, nvar))
9     [1])[1] #Paralelismo
10    tiemposCP <- c(tiemposCP, tcp) #se guarda tiempo
11  }
12 }

```

Se procede a realizar una prueba estadística para comprobar que los resultados del experimento sean significativos. Se elige la prueba de suma de rangos Wilcoxon pues esta prueba no paramétrica permite comparar dos muestras en busca de significancia. El resultado de aplicar esta prueba estadística a las dos muestras que son los vectores *tiemposSP* y *tiemposCP* regresa un valor p el cual es comparado con un valor alfa de significancia.

```

1 #Prueba estadistica
2 wilcox.test(tiemposSP, tiemposCP, alternative = "greater")

```

3. Resultados y conclusiones

En la figura 1 se puede observar como aumenta el tiempo respecto a la cantidad de cúmulos y al número total de partículas.

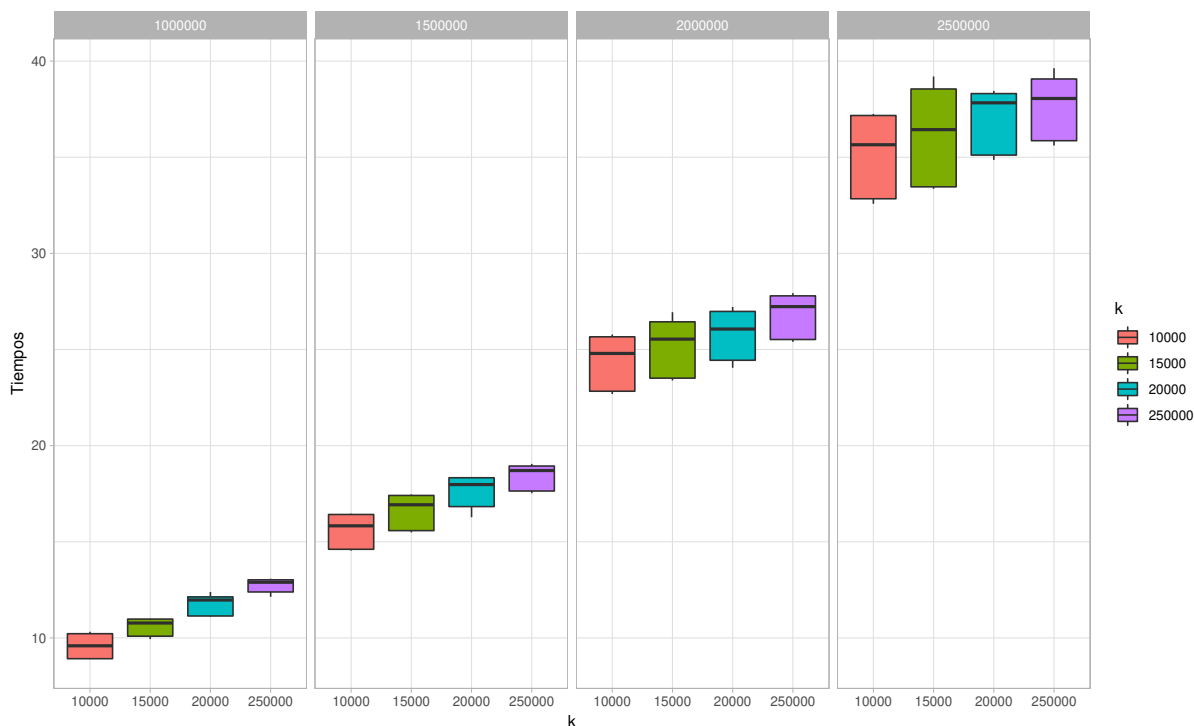


Figura 1: Implementación no paralela de los tiempos de ejecución respecto a k y n

Con esto se puede afirmar que en efecto, si estas dos cantidades incrementan, también el tiempo de ejecución lo hace. En la figura 2 se puede observar como los tiempos de ejecución disminuyen, esto se debe a que esta es la implementación paralela y fue ejecutada usando tres núcleos del procesador.

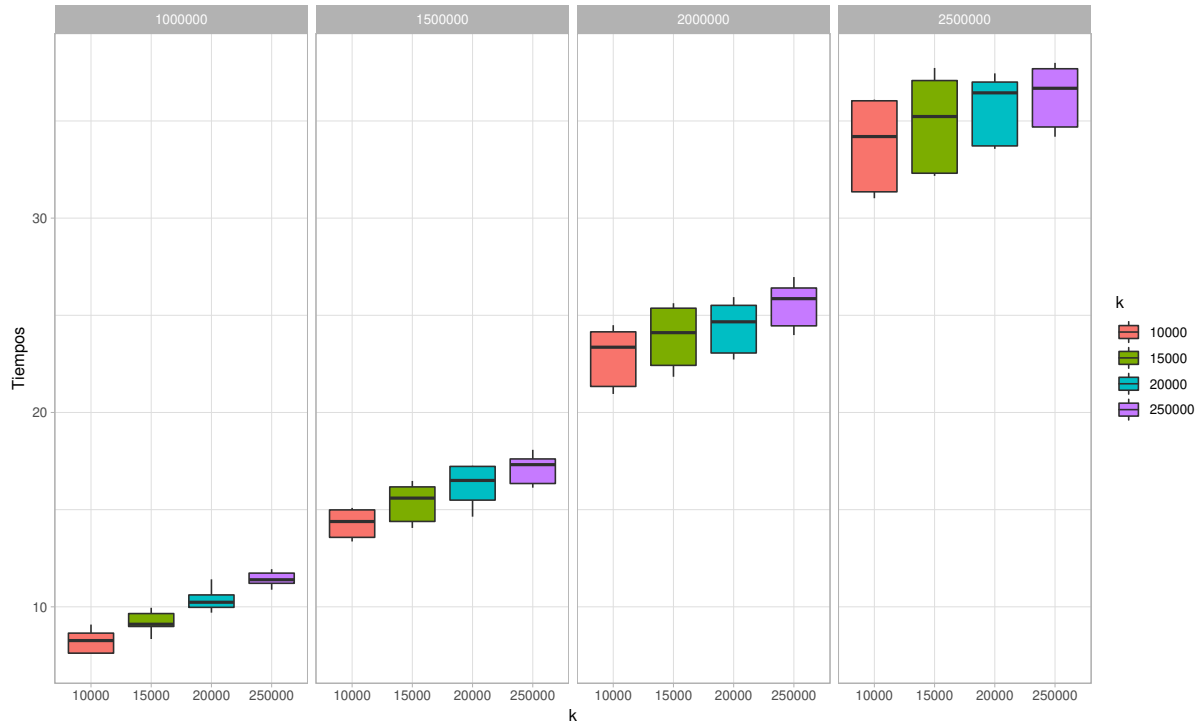


Figura 2: Implementación paralela de los tiempos de ejecución respecto a k y n

Respecto a la prueba estadística de suma de rangos de Wilcoxon, esta regresa un valor p de 0,02833 el cual es menor al valor alfa de significancia de 0,05 y esto nos dice que en efecto, existe una diferencia significativa entre la primera muestra que son los tiempos sin usar el paralelismo respecto a la otra muestra que son los tiempos cuando si se usa el paralelismo. Con una diferencia media de 1.8 segundos en el tiempo de ejecución de ambas implementaciones y con una significancia notable mostrada por la prueba estadística, se afirma que la versión paralela es más rápida que la versión sin usar el paralelismo.

Referencias

- [1] R: R Project, 2019
<https://www.r-project.org/>
- [2] Satu Elisa Schaeffer: Práctica 8: modelo de urnas , 2019
<https://elisa.dyndns-web.com/teaching/comp/par/p8.html>