Tarea 8: modelo de urnas

Simulación de sistemas

2 de octubre de 2017

1. Introducción

En muchas áreas de la física y química, los fenómenos de coalescencia y fragmentación son útiles de analizar. En esta práctica simulamos dichos fenómenos, donde se cuenta con una cantidad de partículas que se unen para formar cúmulos, los cuales pueden fragmentarse en cúmulos de menor tamaño.

2. Especificaciones computacionales

La presente tarea se realizó en una máquina con las siguientes especificaciones: procesador Intel(R)Core(TM) i5-6200U CPU 2.30 GHz 2.40 GHz con 8GB en memoria RAM y sistema operativo Windows 10 Home. Se emplearon tres de los cuatro núcleos.

3. Tarea

En esta tarea se implementa una simulación de modelo de urnas de manera paralela. Además, se realiza una comparación en tiempos de ejecución entre la versión secuencial y la versión paralela.

Para realizar la versión paralela fue necesario crear dos funciones. La primera desarrolla la fase de fragmentación y la segunda desarrolla la fase de unión. Se muestra a continuación las dos funciones mencionadas.

```
faserotura <- function(i){
   urna <- freq[i,]
   if (urna$tam > 1) {
     return(romperse(urna$tam, urna$num))
   } else {
     return(rep(1, urna$num))
   }
 }
```

```
faseunion <- function(i){
   urna <- freq[i,]
   return(unirse(urna$tam, urna$num))
 }
```

Se ejecutaron las versiones secuencial y paralela para comparar los tiempos de ejecución para diez réplicas con una cantidad de partículas $n = 1\,000\,000$ y una cantidad de cúmulos $k = 10\,000$. Los resultados se encuentran la figura 1.

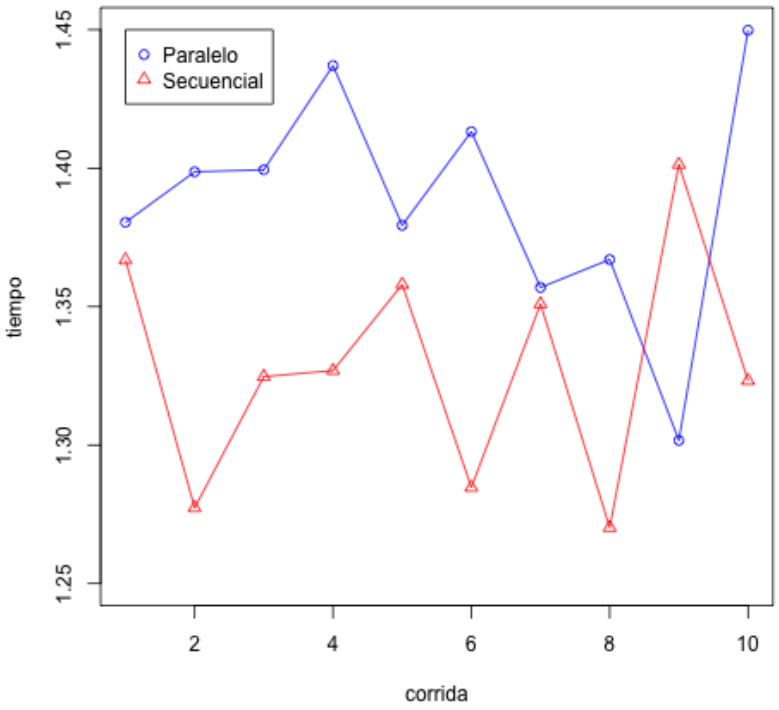


Figura 1: Tiempo de ejecución de diez réplicas con $k = 10\,000$ y $n = 1\,000\,000$.

Observamos en la figura 1 que la manera secuencial tiene menor tiempo de ejecución que la versión paralela cuando $k = 10\,000$ y $n = 1\,000\,000$. La diferencia de tiempos promedio es 0.06 segundos. Para $k = 50\,000$ y $n = 5\,000\,000$ tenemos diferentes resultados los cuales podemos apreciar en la figura 2 y muestran que utilizando la versión paralela se tiene un ahorro de tiempo promedio de 1.04 segundos.

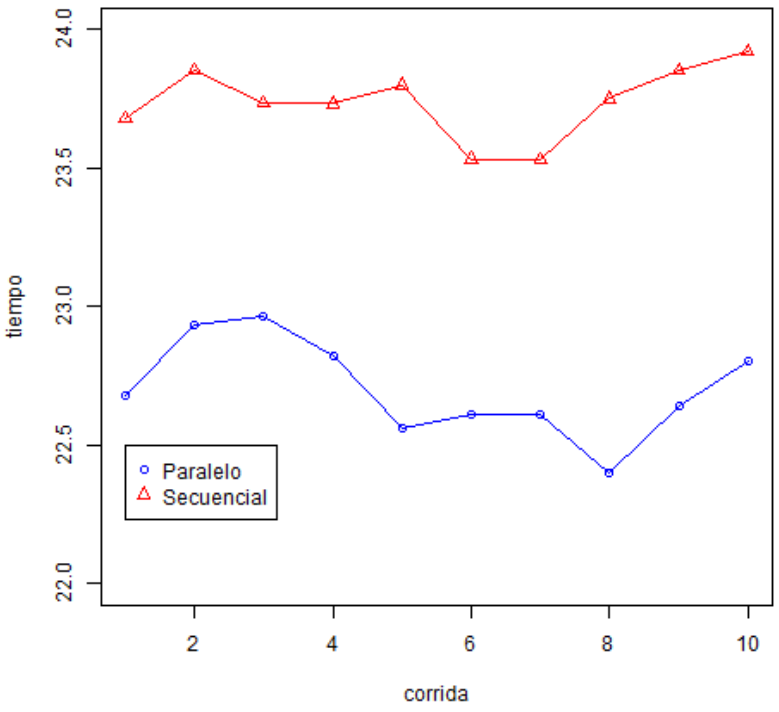


Figura 2: Tiempo de ejecución de diez réplicas con $k = 50\,000$ y $n = 5\,000\,000$.

4. Reto 1

Para el primer reto se pide estudiar el ahorro del tiempo de la versión paralela contra la versión secuencial y analizar si este ahorro es estadísticamente significativo considerando diferentes valores de k, respetando la proporsión $n = 30k$.

Se consideraron seis valores de k distintintos ($1\,000$, $5\,000$, $10\,000$, $15\,000$, $20\,000$, $25\,000$) y para cada uno de ellos se ejecutaron cincuenta réplicas en ambas versiones (paralela y secuencial). Los resultados se muestran en la figura 3 y van intercalados en versión paralela (color azul) y versión secuencial (color rojo) para cada valor de k.

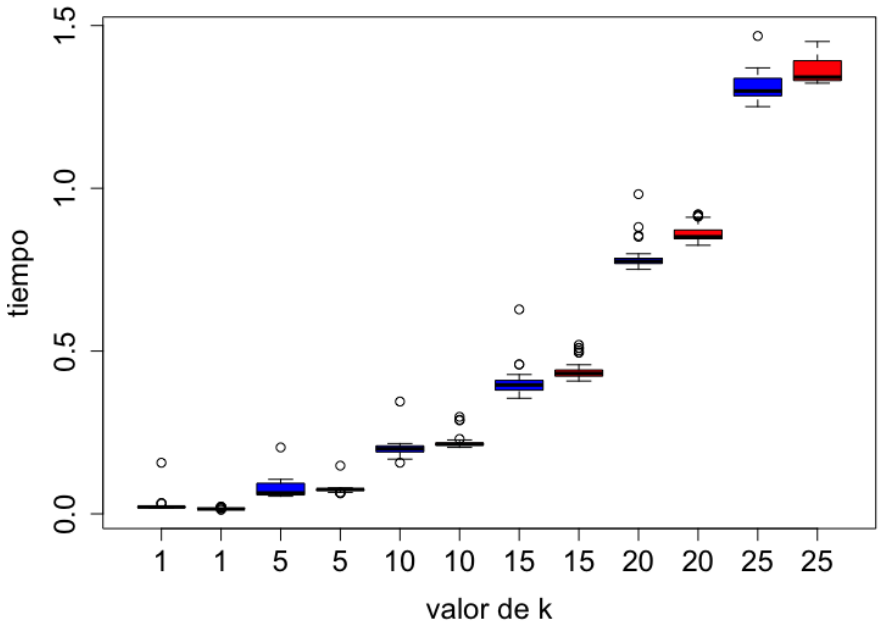
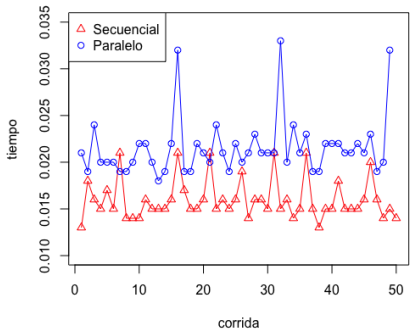


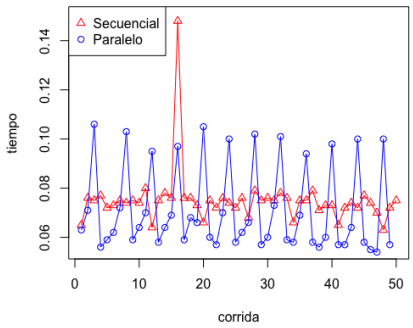
Figura 3: Tiempo de ejecución de cincuenta réplicas para las versiones paralela (azul) y secuencial (rojo) con valores de $k = 1\,000r$ para $r = \{1, 5, 10, 15, 20\}$.

Observando la figura 3 podemos notar que la versión secuencial es más conveniente para valores de k pequeños (1 000 y 5 000), dado que el tiempo de ejecución es menor comparado con la implementación paralela. Para valores de k mayores a 10 000 la versión paralela muestra menores tiempos de ejecución.

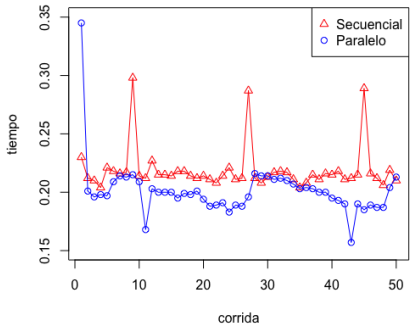
En la figura 4 se muestra los valores obtenidos en la cincuneta replicas para las versiones paralela (azul) y secuencial (rojo) con los diferentes valores de k.



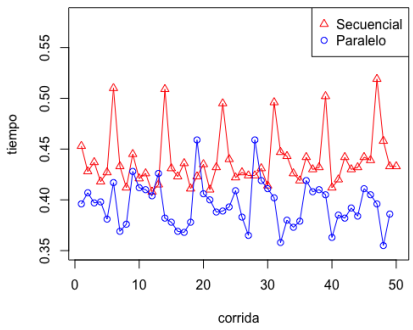
a) $k = 1\,000$



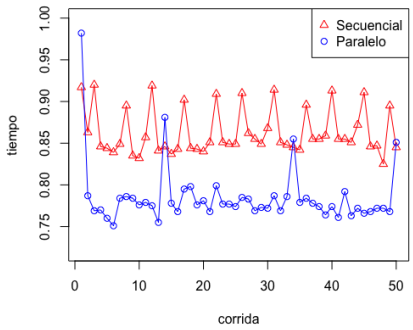
b) $k = 5\,000$



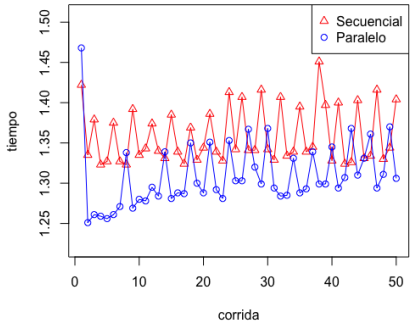
c) $k = 10\,000$



d) $k = 15\,000$



e) $k = 20\,000$



f) $k = 20\,000$

Figura 4: Tiempo de ejecución de cincuenta réplicas para versiones paralela (azul) y secuencial (rojo) con distintos valores de k.

Como vemos en la figura 4 cuando k toma valores de 1 000 o 5 000 la versión secuencial da menores tiempos de ejecución, pero para $k = 10\,000$ la versión paralela es la que da menores tiempos, aunque los valores son muy cercanos a la versión secuencial. A partir de $k = 15\,000$ vemos una diferencia de tiempos mayor entre ambas versiones, teniendo menores tiempos la versión paralelizada.

Ahora, se desea saber si el ahorro en tiempo de ejecución es estadísticamente significativo. Primero comprobaremos si nuestros datos son normales utilizando la prueba de *Shapiro-Wilk*. Los resultados son mostrados en seguida.

```
        Shapiro-Wilk normality test

data:  datosbox
W = 0.8294, p-value < 2.2e-16
```

Como el p-valor es menor a 0,05 se asume la distribución no es normal.

Ahora que sabemos que los datos no son normales la prueba estadística no paramétrica por la que se opto es *Kruskal-Wallis*, cuyos contrastes de hipótesis son los siguientes:

- H_0: los datos provienen de la misma distribución.
- H_1: los datos no provienen de la misma distribución.

Los resultados de la prueba los podemos ver a continuación.

```
        Kruskal-Wallis rank sum test

data:  ambos
Kruskal-Wallis chi-squared = 588.9, df = 11
p-value < 2.2e-16
```

Como podemos observar, el p-valor es menor que 0.05 lo que da lugar a concluir que existen diferencias significativas.