Práctica 3: teoría de colas

Ricardo Rosas Macías

21 de mayo de 2019

1. Introducción

La teoría de colas o líneas de espera, es un análisis matemático que permite estudiar el comportamiento sistemático de un procedimiento que tiene la capacidad de procesar muchos datos o tareas, sin aletargamiento que arruine la obtención de los parámetros de interés.

2. Objetivo

En la práctica se busca examinar el arreglo de las tareas del experimento, obtener los tiempos que tardan en realizarse y su duración al ejecutarse con un número de núcleos determinado.

2.1. Descripción

Lo que se debe hacer es [3]:

> "Examinar cómo las diferencias en los tiempos de ejecución de los diferentes ordenamientos cambian cuando se varía el número de núcleos asignados al cluster, utilizando como datos de entrada un vector que contiene primos grandes; descargados de `https://primes.utm.edu/lists/small/millions/` y no-primos con un mismo número de dígitos. Investiga también el efecto de la proporción de primos y no primos en el vector igual como la magnitud de los números incluidos en el vector con pruebas estadísticas adecuadas."

3. Resultados y conclusiones

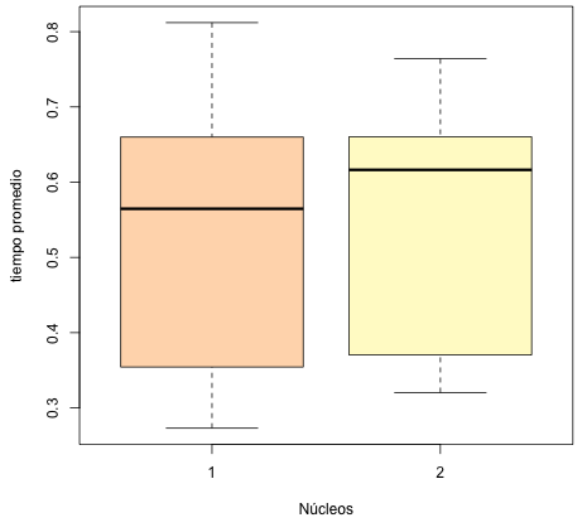
En términos generales para la lograr con el objetivo, se realizaron cambios en el código [1][2] de tal forma que proporcione el comportamiento que tiene la ejecución del experimento con la proporción de primos y no primos.

```
pd <- read.csv("primes1.txt", sep="", header = FALSE) #Primos descargados
pd$V1 <- NULL
pd$V10 <- NULL
```

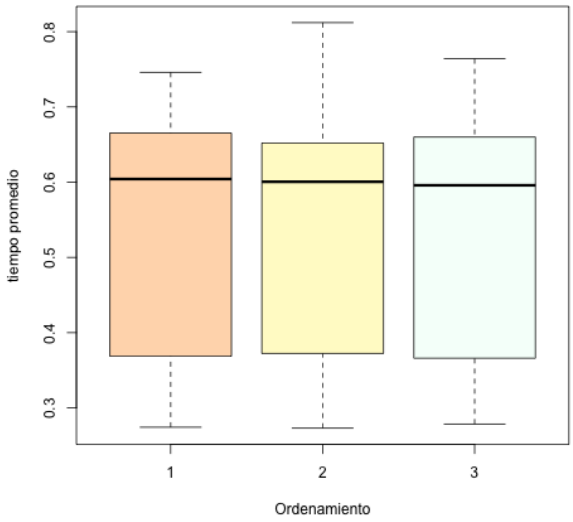
```r
pdm <- as.matrix(pd)
primos <- as.vector(t(pdm))

hastA<-20011
desde<-5381
hasta<-11827
replicas<-30
primo <- function(n) {
  for (i in 2:(n-1)) {
    if ((n > i && n %%i) == 0) { # residuo es cero
      return(FALSE)
    }
  }
  return(n)
}
primos <- numeric() # un vector vacio
for (n in desde:hastA) {
  primos <- c(primos, primo(n)) # combinar vectores
}

noprimo <- function(n) {
  for (i in 2:(n-1)) {
    if ((n > i && n %%i) == 0) { # residuo es cero
      return(n)
    }
  }
  return(FALSE)
}
noprimos <- numeric() # un vector vacio
for (m in desde:hasta) {
  noprimos <- c(noprimos, noprimo(m)) # combinar vectores
}

PC<-primos[which(primos>0)] # Primos crecientes
NPC<-noprimos[which(noprimos>0)] # No primos crecientes
PD<- sort(a,decreasing = TRUE) # Primos decrecientes
NPD<-sort(l,decreasing = TRUE) # No primos decrecientes
PA<-sample(a) # Primos aleatorios
NPA<-sample(d) # No primos aleatorios

e <- sample(a, size=420 ,replace = TRUE, prob = NULL)
f<-sample(l,size=420, replace = TRUE,prob=NULL)

primx <- function(n) {
  if (n == 1 || n == 2) {
    return(TRUE)
  }
  if (n %%2 == 0) {
    return(FALSE)
  }
  for (i in seq(3, max(3, ceiling(sqrt(n))), 2)) {
    if ((n %%i) == 0) {
      return(FALSE)
    }
  }
```

```
  return(TRUE)
}

suppressMessages(library(doParallel))
registerDoParallel(makeCluster(detectCores()-2))
PCt <- numeric()
PDt <-numeric()
PAt <- numeric()
NPDt <-numeric()
NPCt <-numeric()
NPAt <-numeric()

for (r in 1:replicas) {
  PCt <- c(at, system.time(foreach(n = a, .combine=c) %dopar% primx(n))[3])
  PDt <- c(bt, system.time(foreach(n = b, .combine=c) %dopar% primx(n))[3])
  PAt <- c(vt, system.time(foreach(n = v, .combine=c) %dopar% primx(n))[3])
  NPDt <- c(dt, system.time(foreach(n = d, .combine=c) %dopar% primx(n))[3])
  NPCt <- c(lt, system.time(foreach(n = l, .combine=c) %dopar% primx(n))[3])
  NPAt <- c(wt, system.time(foreach(n = w, .combine=c) %dopar% primx(n))[3])
  }
stopImplicitCluster()

mean(PCt)
mean(PDt)
mean(NPDt)
mean(NPCt)
mean(PAt)
mean(NPAt)
nucleos <- rep(1,20)
nucleos<-c(nucleos,rep(2,20))
digitos<-rep(5,10)
digitos<-c(digitos,rep(4,10))
digitos<-c(digitos,rep(3,10))
digitos<-c(digitos,rep(2,10))
digitos<-c(digitos,rep(1,10))
proportion<- rep(1,3)
proportion<-c(proportion,rep(2,3))
proportion<-c(proportion,rep(3,3))
p<-c(1, 2, 3)
ordenamiento<-rep(p, 20)
```

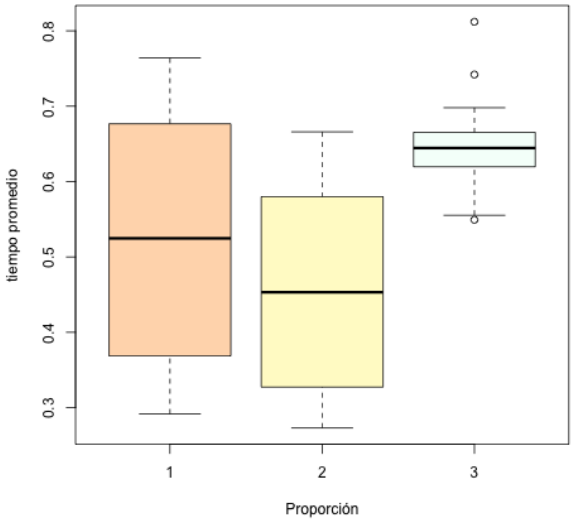
Se realizaron gráficos de la simulación como se muestra en la figura 1, en donde se puede observar en la figura 1(a) el cambio de núcleos para realizar las tareas es ligeramente significativo, la media de tiempo promedio sube un poco pero permanece debajo del segundo cuartil. En cuanto el orden, en la figura 1(b) se muestra que se tarda menos en analizar los datos de manera ascendente; como aparecen en el documento de texto que se descargo de la página web. Sin embargo, en la figura 1(c) muestra una proporción, donde hay algunos cambios en la matriz respecto al tiempo del ciclo. Finalmente podemos observar en la figura 1(d) el número de dígitos en el cual se distingue claramente que se tarda mucho más al realizar el análisis de números de 2 dígitos.



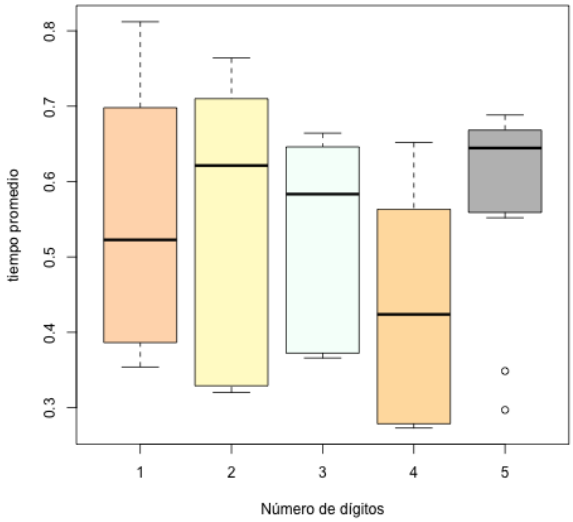
(a) Núcleos



(b) Orden



(c) Proporción



(d) Dígitos

Figura 1: Resultados de ejecución del experimento

Referencias

[1] Ricardo Rosas Macías. Práctica 3: teoría de colas, 2019. URL `https://github.com/RicardoRosMac/Simulation/tree/master/HWP3`.

[2] Liliana Saus. p3 teoría de colas, 2018. URL `https://github.com/pejli/simulacion/tree/master/P3`.

[3] Elisa Schaeffer. Práctica 3: teoría de colas, 2019. URL `https://elisa.dyndns-web.com/teaching/comp/par/p3.html`.