

## Práctica 3: Teoría de colas

Gabriela Sánchez Y.

En esta práctica se estudia el efecto que tienen el orden de ejecución y el número de núcleos utilizados para resolver un trabajo, en el tiempo de ejecución del mismo. Como trabajo ejemplo se utiliza la examinación de si un número entero dado es o no un número primo.

La tarea que se realiza en el experimento es determinar si los números en un determinado rango (1000-3000) son primos o no, ordenando los números correspondientes en diferentes formas: orden normal, orden inverso y orden aleatorio (el mismo en cada repetición). El experimento se repite un total de 10 ocasiones para cada número de núcleos asignados.

Para estudiar el efecto que tiene el número de núcleos asignados en el tiempo de ejecución, se modificó el código base de manera que este varíe el número de núcleos. En este caso la simulación se realizó en una computadora de cuatro núcleos de los cuales solamente se utilizaron tres. Esta operación se realiza con un ciclo que se encarga de variar la asignación:

```
for (nu in 1:nucleos) {  
  cluster <- makeCluster(nu)  
  registerDoParallel(cluster)  
  ot <- numeric()  
  it <- numeric()  
  at <- numeric()  
  for (r in 1:replicas) {  
    ot <- c(ot, system.time(foreach(n = original, .combine=c)  
      %dopar% primo(n))[3]) # de menor a mayor  
    it <- c(it, system.time(foreach(n = invertido, .combine=c)  
      %dopar% primo(n))[3]) # de mayor a menor  
    at <- c(at, system.time(foreach(n = aleatorio, .combine=c)  
      %dopar% primo(n))[3]) # orden aleatorio  
  }  
  
  resultadosOT <- rbind(resultadosOT,ot) #guarda resultados para  
  #el orden original
```

```

dataOT <- matrix(t(resultadosOT), nrow=replicas, ncol=nucleos)

resultadosIT <- rbind(resultadosIT,it) #para el orden inverso
dataIT <- matrix(t(resultadosIT), nrow=replicas, ncol=nucleos)

resultadosAT <- rbind(resultadosAT,at) #para el orden aleatorio
dataAT <- matrix(t(resultadosAT), nrow=replicas, ncol=nucleos)

stopImplicitCluster()
stopCluster(cluster)
}

```

En la figura 1 se grafican en un diagrama caja-bigote los resultados obtenidos de la simulación. La primera gráfica muestra los tiempos de ejecución para determinar los números primos en el intervalo de prueba cuando se encuentra en el orden original, la gráfica dos y tres muestran los resultados para el orden inverso y aleatorio respectivamente. Los tiempos son medidos en segundos. Se observa que hay una variación en los tiempos de ejecución si se varía el número de núcleos asignados a la tarea.

El efecto que tiene la asignación de núcleos a la tarea sobre el tiempo de ejecución del trabajo se debe a que mientras menos núcleos se tengan más "trabajo" deben realizar los que sí están asignados.

Se piensa que además del número de núcleos, el orden y el tipo de los datos para la tarea también influye en el tiempo de ejecución. Por ejemplo, el tiempo que tardaría en determinar cuales de los primeros 100 enteros son primos no es el mismo que tener como datos de entrada 100 números primos, porque deben realizarse más operaciones. Más aún, si en lugar de los primeros 100 primos utilizo 100 primos relativamente grandes, el trabajo sería mayor por la misma razón, se deben realizar más operaciones.

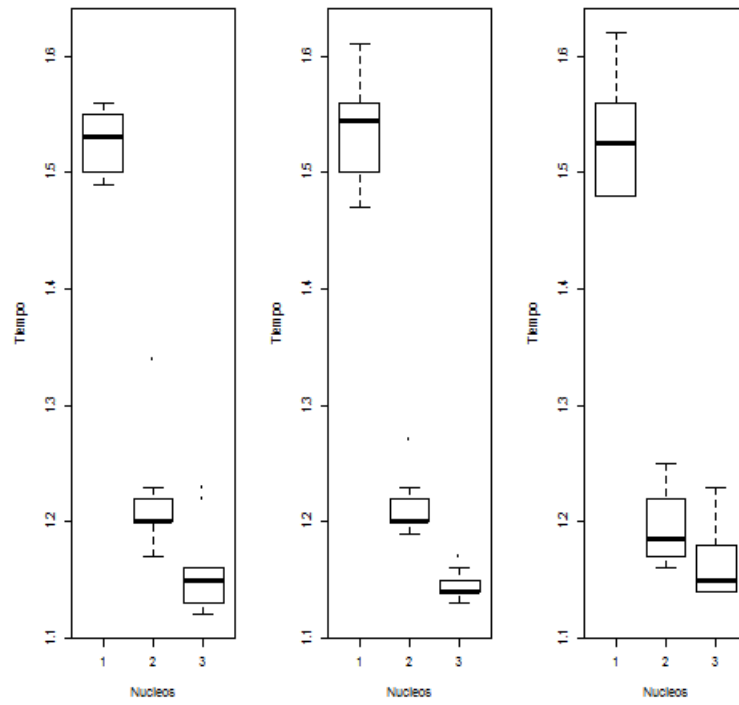


Figura 1: Variaciones en los tiempos de ejecución de tareas respecto al número de núcleos asignados