

Practica 3: Teoría de Colas

29 de agosto de 2017

Resumen

En esta práctica el interés será comparar los tiempos de ejecución de diversos modos de ordenamiento de números ya establecidos, variando la cantidad de núcleos del ordenador utilizados para la misma.

1. Introducción

La tarea en específico consistirá en saber si un número es primo o no, bien podemos dar una lista de números ya establecida para checar o podemos pedirle que compruebe de un cierto número a otro si lo es. La teoría de Colas entra en nuestra experimentación cuando observamos que para checar si un número en efecto es primo tarda diferente dependiendo si el número es muy grande o muy pequeño, de aquí que lo pudiéramos visualizar como si tuviéramos una cantidad de números por analizar, y varias “máquinas” o líneas para procesarlo, en este caso nuestros núcleos.

Hasta este punto podríamos suponer que si tengo menos núcleos los tiempos en “línea” serán más grandes, pero veamos si esto ocurre en la práctica, tendremos 3 formas diferentes de checar los números deseados, de manera ascendente, descendente y de forma aleatoria.

2. Parámetros de Trabajo

Para llevar a cabo la experimentación se realizó en una MacBook Pro la cual cuenta con 4 núcleos disponibles, como es necesario disponer de uno de los núcleos para las ejecuciones, la experimentación la realizaremos desde uno hasta tres núcleos, de igual forma ordenaremos los números con un número de repeticiones fijo, en nuestro caso serán 20 repeticiones, de las cuales se comprobarán números desde mil hasta tres mil.

3. Modificaciones del código

Para la primer parte de la tarea lo único que fue necesario incluir fue la automatización para cambiar la cantidad de núcleos utilizados, así como la generación de las gráficas para cada una de las corridas con los núcleos, para esto se agregó lo siguiente al código:

```
for(num in 1: (detectCores()-1)){  
  registerDoParallel(makeCluster(num))  
  .  
  .  
  .  
  salida = paste("p3_t", num, ".png", sep="")  
  png(salida)  
  boxplot(ot, it, at)  
  graphics.off()  
}
```

4. Resultados

A continuación presentaremos los gráficos variando la cantidad de núcleos como se especifica de cada uno en el pie del gráfico.

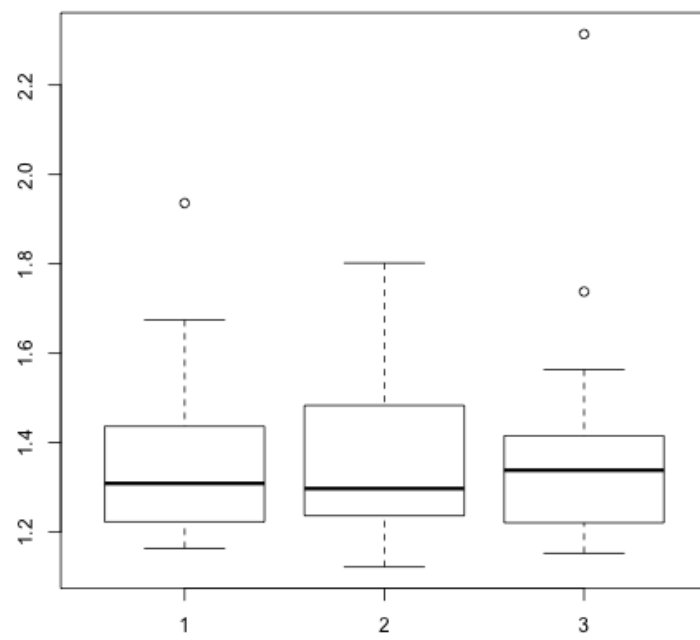


Figura 1: 3 Núcleo

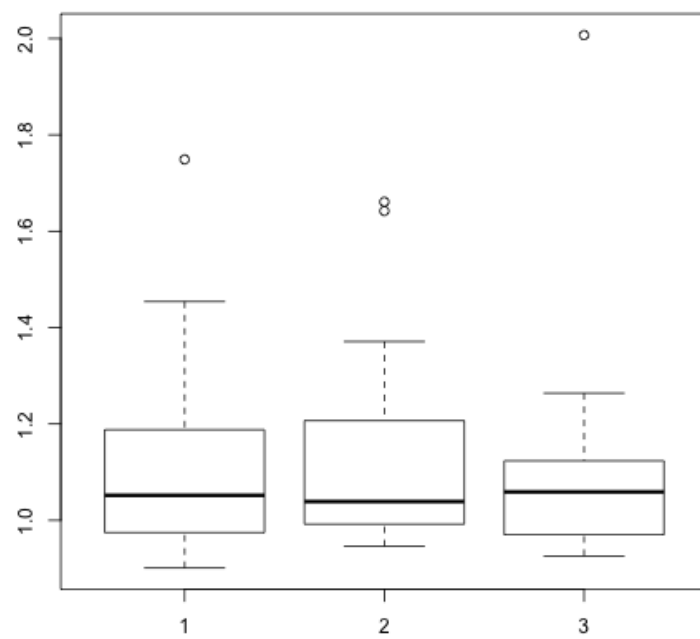


Figura 2: 2 Núcleo

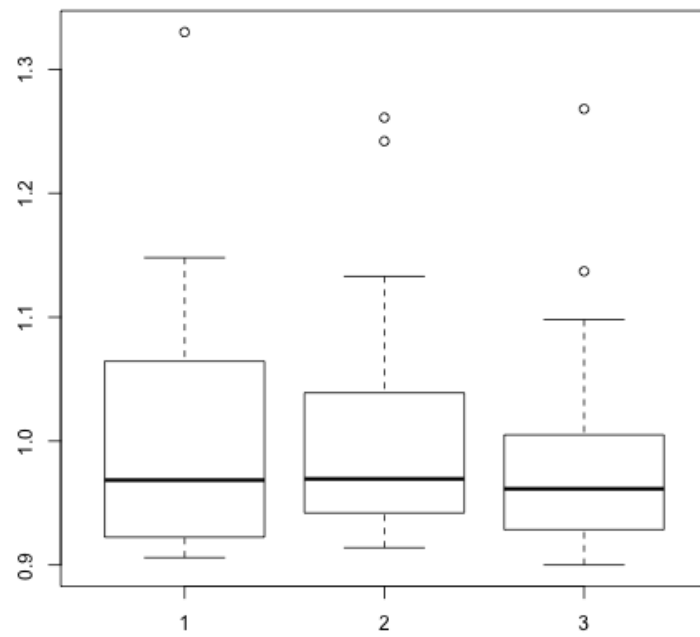


Figura 3: 1 Núcleo

4.1. Interpretación