

Práctica 3

Teoría de colas

Introducción

En matemáticas, existe cierta área que se dedica a estudiar el comportamiento de las líneas de espera dentro de un sistema. Una de las aplicaciones de esta teoría es que los núcleos de una computadora, que ejecutan procesos mediante líneas de espera, y tienen un tiempo total de ejecución. Cada línea de espera tiene un comportamiento diferente y realiza cierta tarea de distinta manera.

En este reporte se estudiará el efecto del orden de ejecución de una tarea, que es la de ordenar vectores, que en este caso contienen números primos) de manera ordenada (menor a mayor), orden invertido (mayor a menor) y orden aleatorio.

Hipótesis

El número de núcleos dedicados a una tarea en específico afecta al tiempo que se tarda en realizarla, es decir, entre menos núcleos se dediquen a realizar esta tarea, más tiempo le tomará realizarla.

Simulación

Se inició un programa para detectar los números primos desde 1000 hasta 3000, y ordenar los números en tres tipos de ordenamiento, de menor a mayor, de mayor a menor y aleatoriamente.

Cada tipo de ordenamiento tomó su tiempo de ejecución en el sistema, y para poder saber la diferencia entre el tiempo que le toma hacer cada tipo de ordenamiento, se usa la función de *system.time*, y hacerlo para cada tipo de ordenamiento, por cada vez que se replique este proceso.

Esto es con un número de núcleos fijo, pero no hace un recorrido en dedicar cierto número de núcleos para hacer el proceso, así que se hizo un ciclo para ir registrando el tiempo que le tomaba hacer la tarea dependiendo del número de núcleos dedicados a hacer el programa. Primero se empezó con uno, y se aumentó

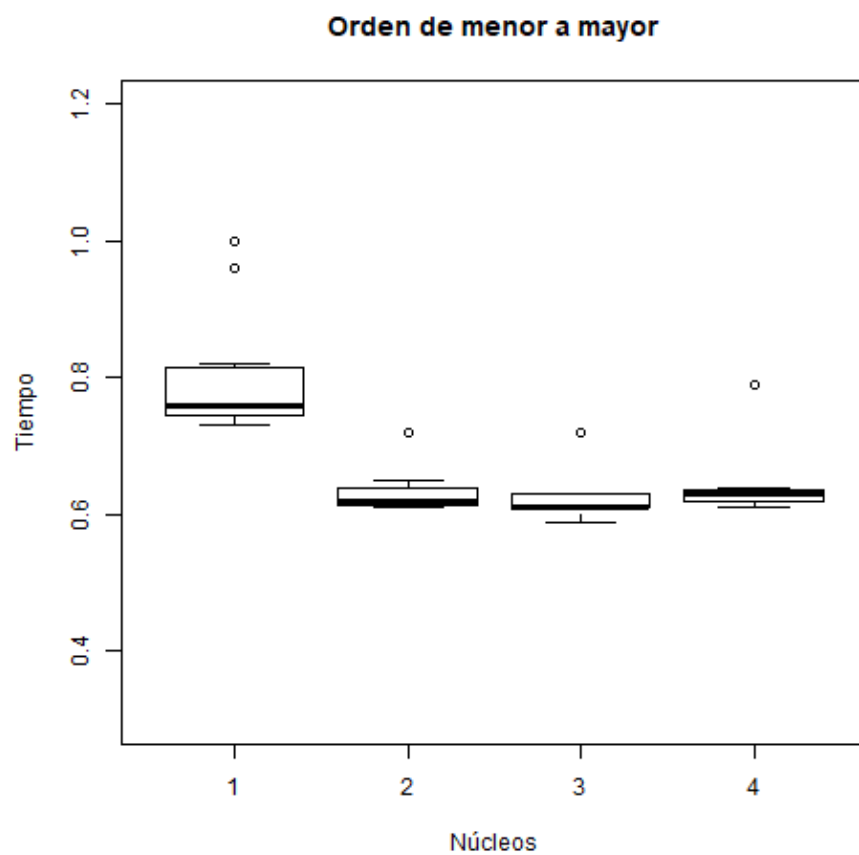
de uno en uno, hasta llegar al número de núcleos totales en la computadora, que en este caso fueron cuatro.

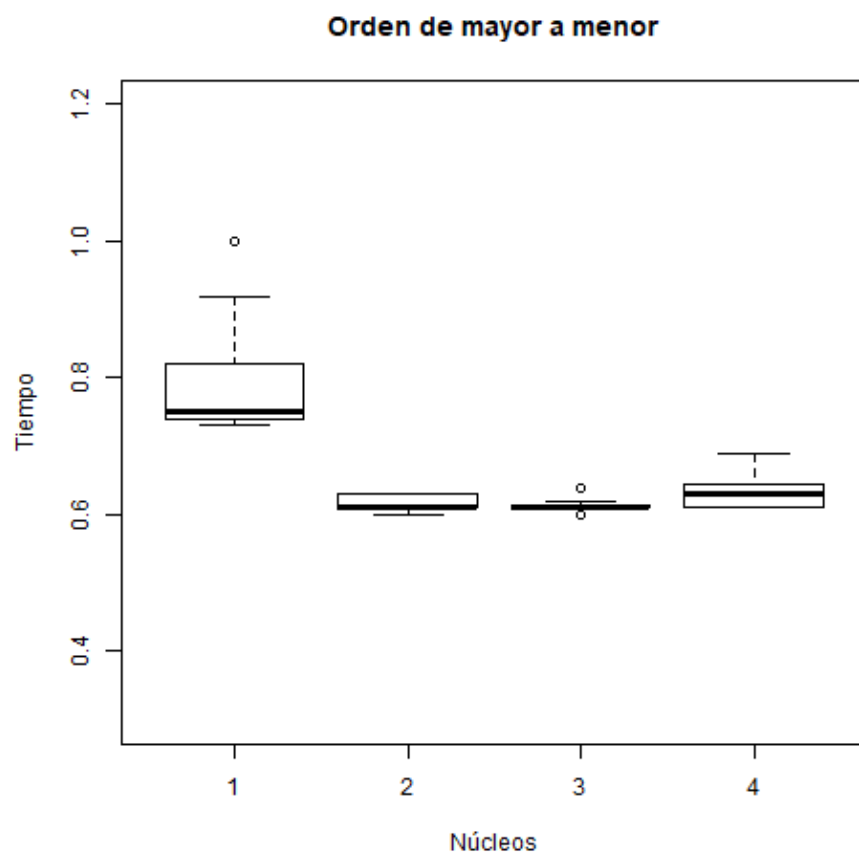
Finalmente, se registraron los resultados de los tiempos en una gráfica, cada una por tipo de ordenamiento, y el tiempo en segundos que le tomaba a los núcleos realizar este programa.

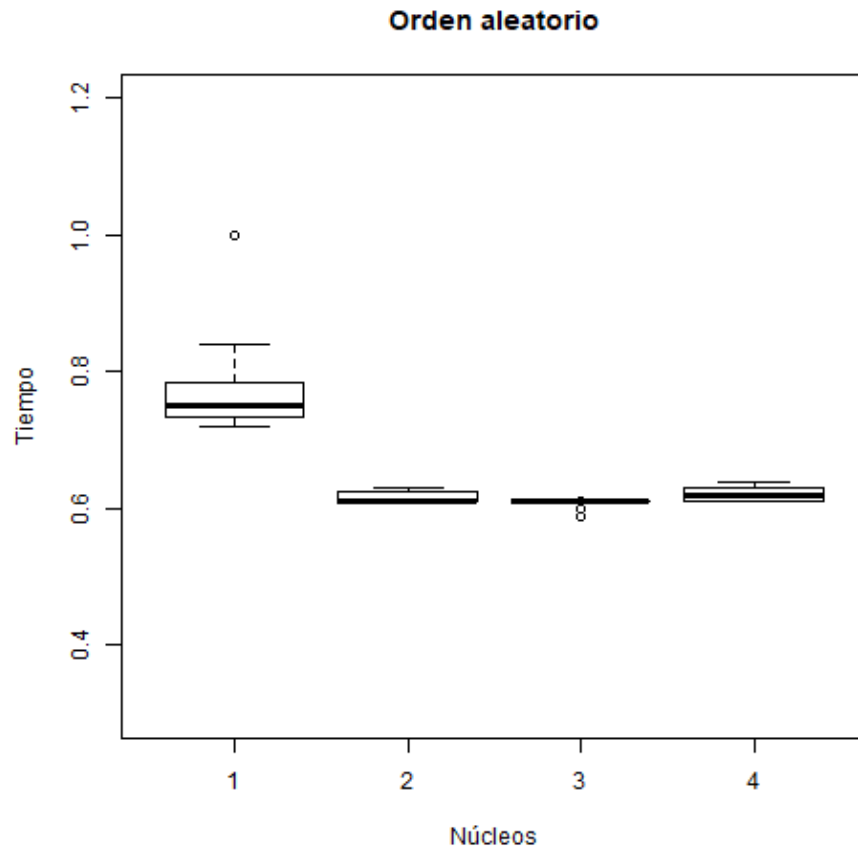
Resultados

Al realizar la primera simulación, de ver solo el tiempo que se tardaba en ejecutar la tarea dedicando los núcleos totales de la computadora menos uno, se notó que hay una diferencia de tiempo dedicado a hacer las tareas dependiendo del ordenamiento. El ordenamiento de menor a mayor tomó más tiempo para hacerse, mientras que el ordenamiento aleatorio fue el que tomó menos tiempo, comparando con los otros tipos de ordenamiento.

Cuando se toma en cuenta el número de núcleos dedicados a realizar el proceso de ordenamiento, el resultado fue que cuando se dedica un núcleo, se tarda más en realizar esta tarea que cuando se dedican más núcleos a realizar estos procesos, siendo de más de un segundo en ejecutar los procesos de ordenamiento cuando se utiliza un núcleo para el proceso, pero dio el caso que cuando se usaron todos los núcleos, tomaba más tiempo hacer el proceso comparando a cuando se utilizan dos o tres núcleos, pero aun así, tomó menos tiempo que cuando se dedicaba un solo núcleo a ejecutar esta tarea.







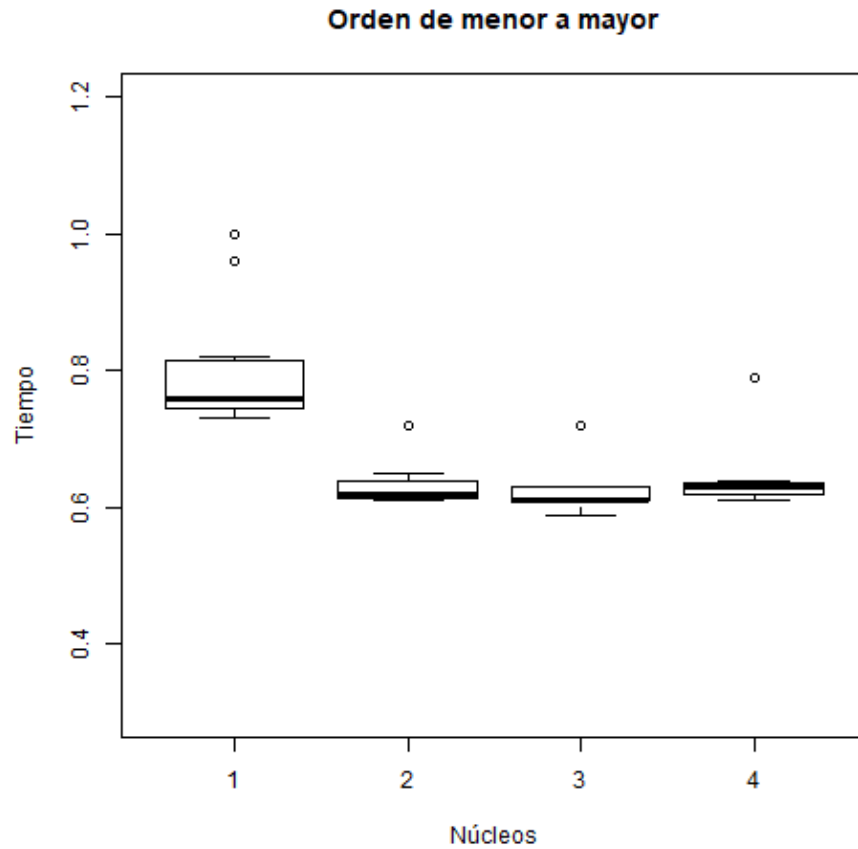
Análisis de tiempos

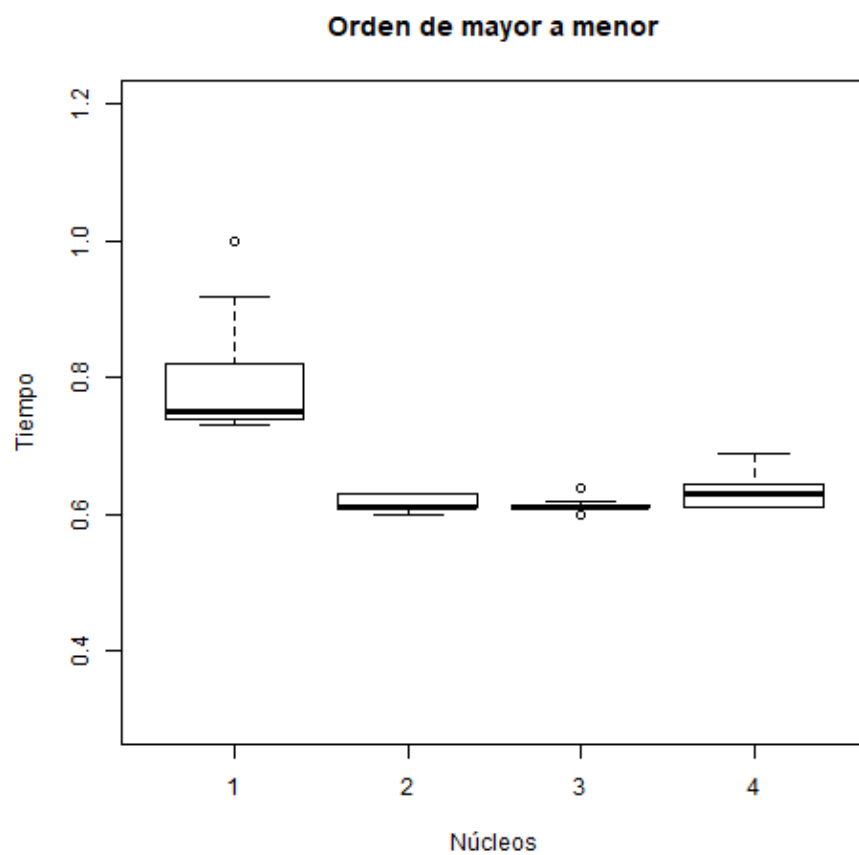
Primero vamos a analizar por qué se tarda más tiempo en hacer el ordenamiento de mayor a menor y por que tarda menos en hacer el ordenamiento aleatorio. Esto se debe a que cuando se ejecuta una tarea y ya está hecha, es más fácil desordenarla, y como el ordenamiento aleatorio se basa en el ordenamiento de mayor a menor, y para cuando se hará el aleatorio, ya está hecho el ordenamiento, por lo tanto, no le toma tanto tiempo en ejecutar esta tarea.

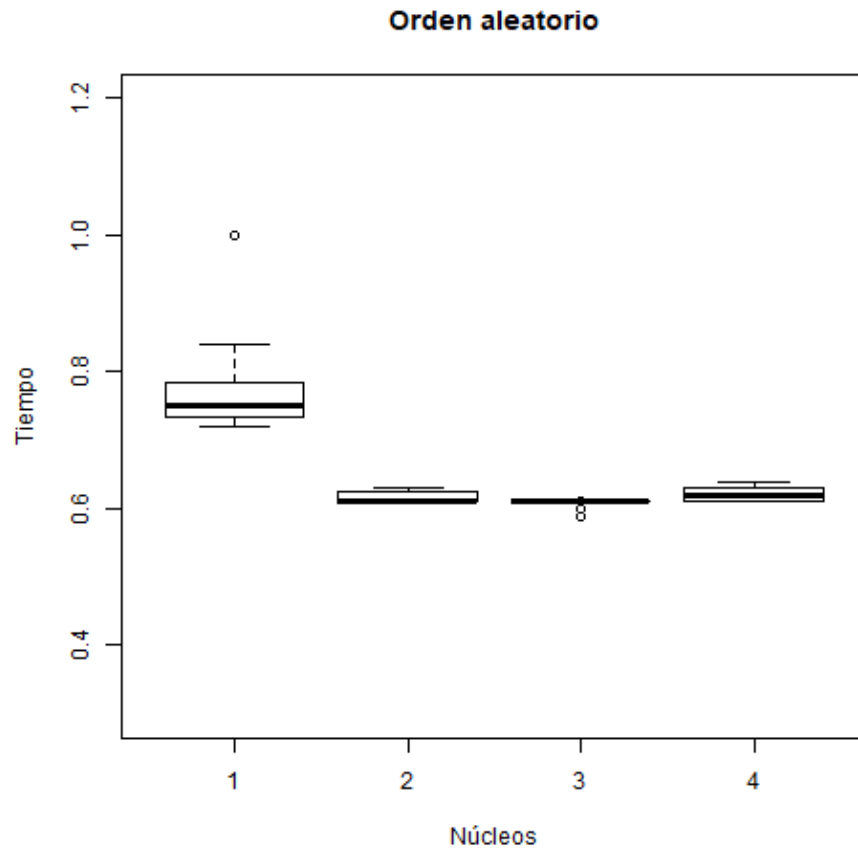
Ahora, las posibles causas por las que haya diferencias en los núcleos dedicados es por la cantidad. Cuando se usa un núcleo, la tarea se vuelve muy pesada para la computadora, y como solo hay uno dedicado al proceso, le cuesta más tiempo terminarlo. Cuando tiene más núcleos dedicados, le cuesta menos tiempo terminarlo, pero el rendimiento puede ir disminuyendo por la cantidad de procesos que ya ha hecho previamente, y puede que se vayan acumulando y le cueste más

trabajo, simplemente por haber ejecutado ya una tarea muy pesada, y cueste mucho rendimiento a los procesadores que quedan.

En las gráficas se observa que le toma un poco más de tiempo cuando se dedican cuatro núcleos al proceso que cuando se dedican dos o tres.







Conclusiones

El número de núcleos dedicados a hacer ciertas tareas afecta en el tiempo que se dedica a terminirlas, y el tipo de tarea afecta también al tiempo, pero si se hace una tarea basada en una que ya se hizo previamente, es posible que se ejecute más rápido que la tarea que se había terminado previamente.