

Infraestructura computacional

Reporte Caso 3

Mateo Perez 201814723

Identificación de la plataforma.

Defina la máquina en la que correrá la aplicación e identifique las siguientes características. Tenga en cuenta que su aplicación deberá correr en la misma máquina para todos los experimentos (si cambia de máquina, los resultados no serán comparables).

- Arquitectura: *64 bits*
- Número de núcleos: *4 cores*
- Velocidad del procesador: *4.00 GHZ*
- Tamaño de la memoria RAM: *16GB*
- Espacio de memoria asignado a la JVM : *2048MB*

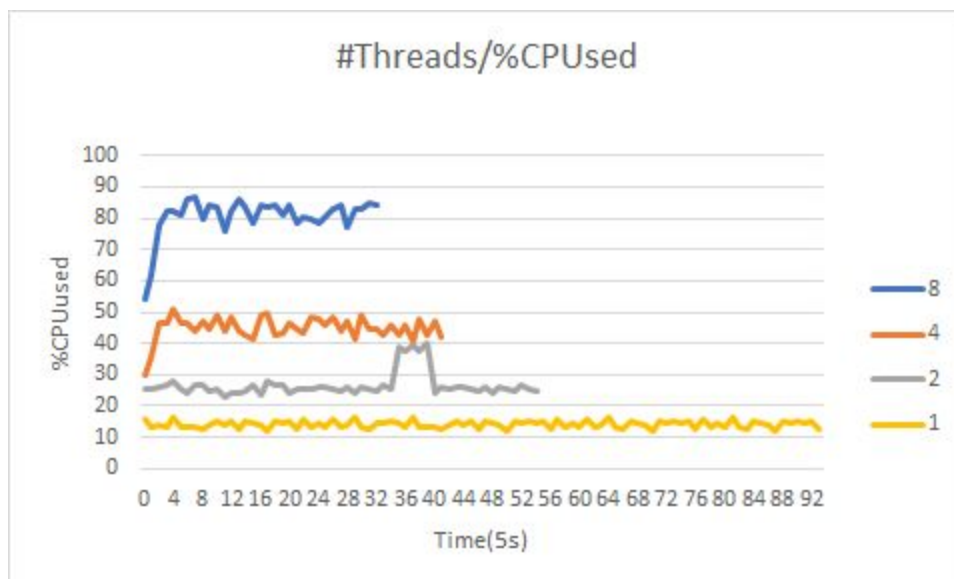
Comportamiento de la aplicación.

Con diferentes estructuras de administración de la concurrencia. Este escenario tiene como objetivo evaluar cambios en el comportamiento de la aplicación ante diferencias en la carga:

Escenarios:

MD5

#Threads/%CPU



En este gráfico se comparó el uso del procesador CPU en función del número de threads. Se evidencia una clara tendencia de aumento de uso de CPU al aumentar el número de threads. Para el caso de 1 thread el uso oscila entre 15%. Con 2 threads el uso es de 25% aproximadamente. Además, resaltar un dato atípico en que el uso aumenta considerablemente hasta 40% sin embargo se vuelve a mantener la tendencia de uso previa. Con 4 threads evidenciamos un uso aproximado del 45% del procesador. Por último, el caso en el que se emplean los 8 threads el procesador llega a ser ocupado en un 85%. Resaltar que en este punto la máquina comenzó a presentar sobrecalentamiento.

#Threads/Tiempo

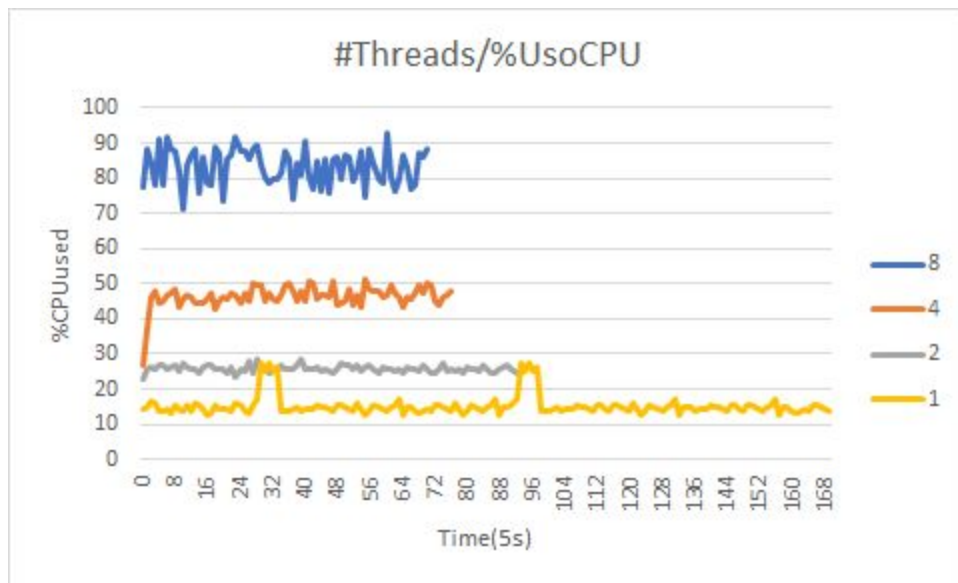


En este gráfico se se comparó el tiempo de ejecución en función del número de threads.

Se puede evidenciar una reducción en el tiempo de ejecución siempre que hay un número mayor de threads. Al pasar de un thread a dos se redujo el tiempo casi en en 50%. Sin embargo, este comportamiento no se repite al duplicar los threads en 4 u 8. Pues cada vez la reducción del tiempo es menor con respecto al aumento de threads anterior.

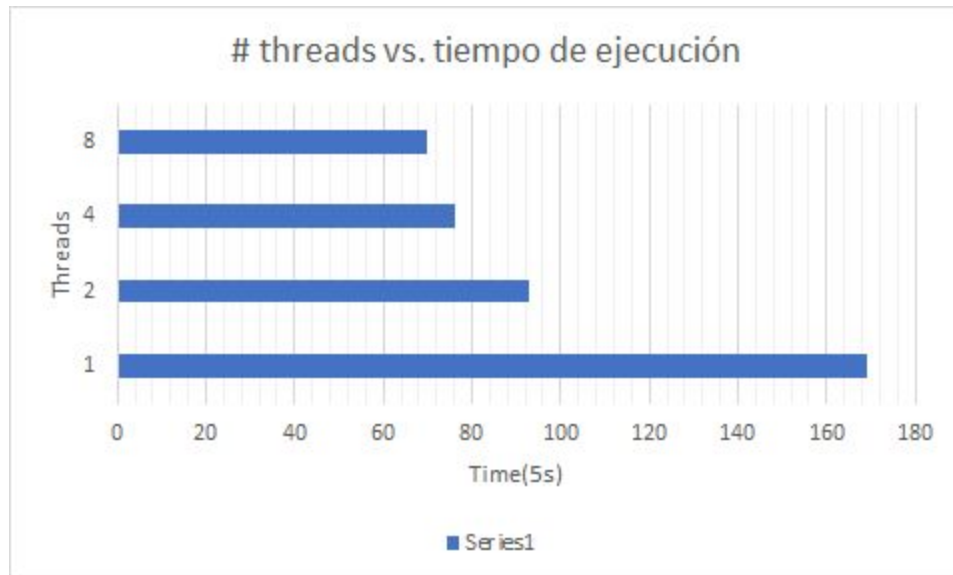
SHA256

#Threads/%CPU



En este gráfico se comparó el uso del procesador CPU en función del número de threads. Se evidencia una clara tendencia de aumento de uso de CPU al aumentar el número de threads. Para el caso de 1 thread el uso oscila entre 15%. Además, resaltar dos casos atípicos en que el uso aumenta considerablemente hasta 25%. Sin embargo, se vuelve a mantener la tendencia de uso previa. Con 2 threads el uso es de 25% aproximadamente. Con 4 threads evidenciamos un uso aproximado del 45% del procesador. Por último, el caso en el que se emplean los 8 threads el procesador llega a ser ocupado en un 85%. Resaltar que en este punto la máquina comenzó a presentar sobrecalentamiento.

#Threads/Tiempo

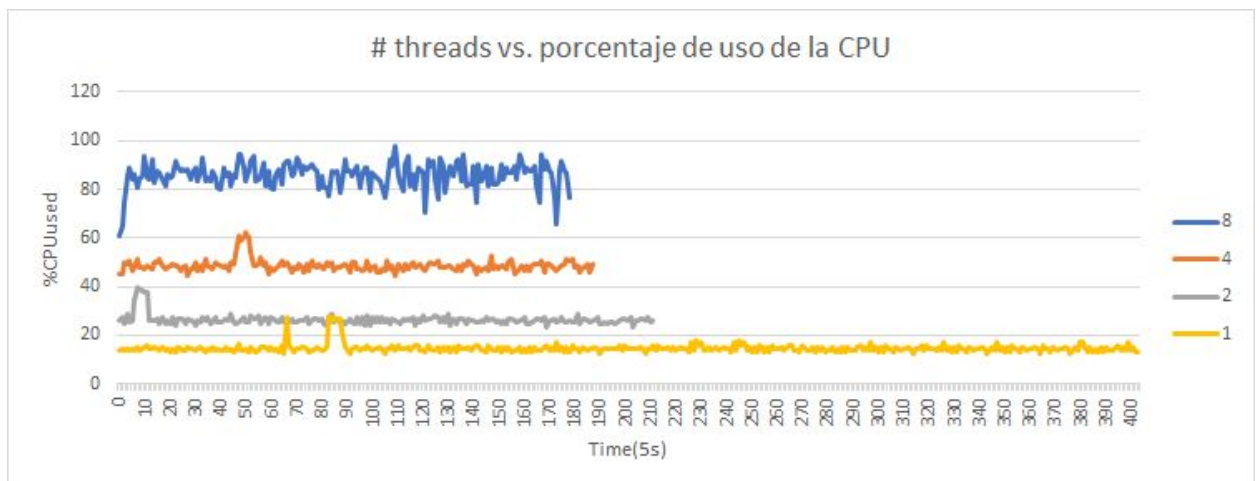


En este gráfico se se comparó el tiempo de ejecución en función del número de threads.

Se puede evidenciar una reducción en el tiempo de ejecución siempre que hay un número mayor de threads. Al pasar de un thread a dos se redujo el tiempo casi en 50%. Sin embargo, este comportamiento no se repite al duplicar los threads en 4 u 8. Pues cada vez la reducción del tiempo es menor con respecto al aumento de threads anterior.

SHA512

#Threads/%CPU



En este gráfico se comparó el uso del procesador CPU en función del número de threads. Se evidencia una clara tendencia de aumento de uso de CPU al aumentar el número de threads. Para el caso de 1 thread el uso oscila entre 18%. Además, resaltar dos casos atípicos en que el uso aumenta considerablemente hasta 25%. Sin embargo, se vuelve a mantener la tendencia de uso previa. Con 2 threads el uso es de 25% aproximadamente, resaltar un caso atípico en que el uso aumenta considerablemente hasta 40%. Con 4 threads evidenciamos un uso aproximado del 45% del procesador. Resaltar un caso atípico en que el uso aumenta considerablemente hasta más del 60%. Por último, el caso en el que se emplean los 8 threads el procesador llega a ser ocupado en un 85%. Resaltar que en este punto la máquina comenzó a presentar sobrecalentamiento. Además de que fue precisamente este el caso en el que más se demoró en descriptar.

#Threads/Tiempo



En este gráfico se se comparó el tiempo de ejecución en función del número de threads.

Se puede evidenciar una reducción en el tiempo de ejecución siempre que hay un número mayor de threads. Al pasar de un thread a dos se redujo el tiempo casi en en 50%. Sin embargo, este comportamiento no se repite al duplicar los threads en 4 u 8. Pues cada vez la reducción del tiempo es menor con respecto al aumento de threads anterior.

