08/09/2018

Jonathan Alexander Prieto Urquijo
2103258

Introducción al paralelismo - hilos

Parte I Hilos Java

- De acuerdo con lo revisado en las lecturas, complete las clases CountThread, para que las mismas definan el ciclo de vida de un hilo que imprima por pantalla los números entre A y B.

- Complete el método main de la clase CountMainThreads para que:
    - Cree 3 hilos de tipo CountThread, asignándole al primero el intervalo [0..99], al segundo [99..199], y al tercero [200..299].
    - Inicie los tres hilos con 'start()'.
    - Ejecute y revise la salida por pantalla.
    - Cambie el inicio con 'start()' por 'run()'. Cómo cambia la salida?, por qué?.
        - Los números aparecen en secuencia. Porque no se están usando los hilos i.e., se está llamando al método que se implementó en la clase CountThread.

Parte II Hilos Java

Parte II.I Para discutir el Martes (NO para implementar aún)

La estrategia de paralelismo antes implementada es ineficiente en ciertos casos, pues la búsqueda se sigue realizando aún cuando los N hilos (en su conjunto) ya hayan encontrado el número mínimo de ocurrencias requeridas para reportar al servidor como malicioso. Cómo se podría modificar la implementación para minimizar el número de consultas en estos casos?, qué elemento nuevo traería esto al problema?

- Crearía una variable a la que cada thread acceda para reportar las ocurrencias encontradas. Una vez ésta sea igual a 5, detener el ciclo de búsqueda que se implementa en la clase SearchThread.java. Por lo visto en clase se podría crear dicha variable como atomica para así evitar una condición de carrera.

Parte III Evaluación de Desempeño

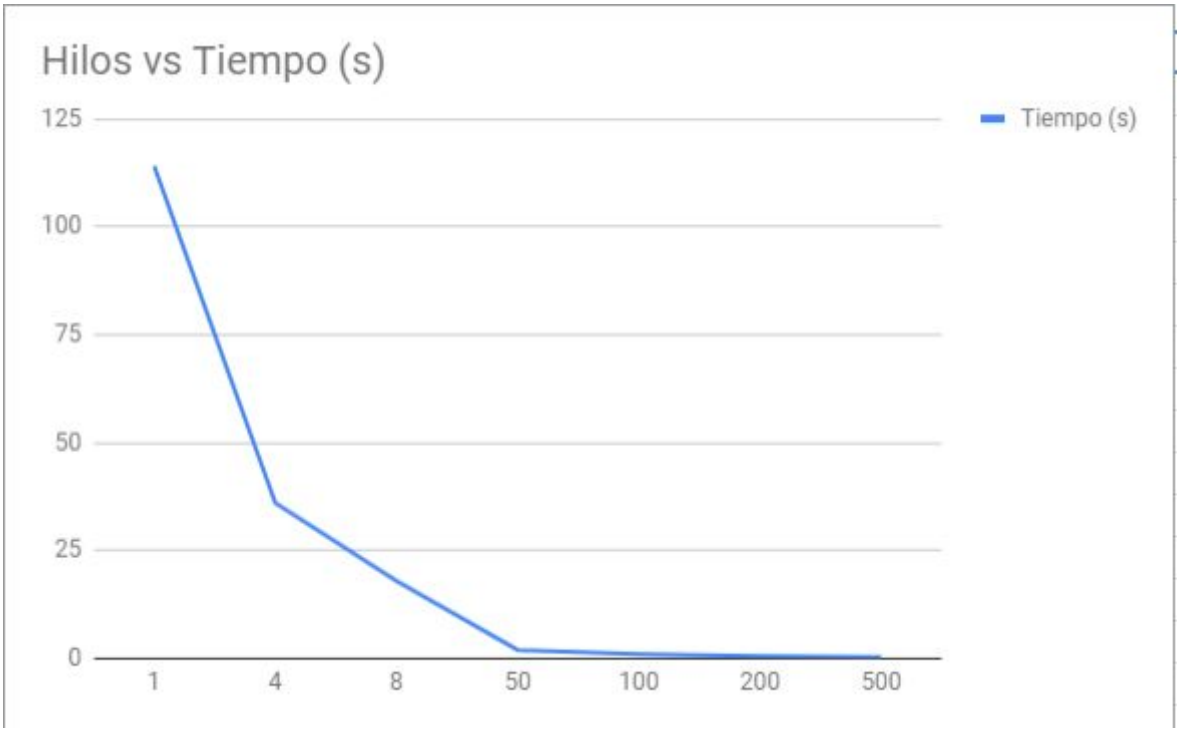
A partir de lo anterior, implemente la siguiente secuencia de experimentos para realizar las validación de direcciones IP dispersas (por ejemplo 202.24.34.55), tomando los tiempos de ejecución de los mismos (asegúrese de hacerlos en la misma máquina):

- Un solo hilo.
- Tantos hilos como núcleos de procesamiento (haga que el programa determine esto haciendo uso del API Runtime).
- Tantos hilos como el doble de núcleos de procesamiento.
- 50 hilos.
- 100 hilos.

	Hilos	Tiempo (s)	CPU	Memoria (B)
	1	114	-131,60%	28.633.088
Núcleos de procesamiento	4	36	0,00%	20.453.232
Doble de núcleos	8	18	0,70%	18.405.440
	50	2	unknown	23.607.552
	100	1	unknown	50.574.976
	200	0,576	unknown	unknown
	500	0,376	unknown	unknown

Al iniciar el programa ejecute el monitor jVisualVM, y a medida que corran las pruebas, revise y anote el consumo de CPU y de memoria en cada caso.

Con lo anterior, y con los tiempos de ejecución dados, haga una gráfica de tiempo de solución vs. número de hilos. Analice y plantee hipótesis con su compañero para las siguientes preguntas (puede tener en cuenta lo reportado por jVisualVM):



Según la ley de Amdahls:

1. $$S(n) = \frac{1}{(1-P) + \frac{P}{n}}$$, donde S(n) es el mejoramiento teórico del desempeño, P la fracción paralelizable del algoritmo, y n el número de hilos, a mayor n, mayor debería ser dicha mejora. Por qué el mejor desempeño no se logra con los 500 hilos?, cómo se compara este desempeño cuando se usan 200?.

    a. Con base en el gráfico se puede ver una cota en t=0, lo cual se puede interpretar como una indiferencia en la cantidad de hilos i.e., llega un punto en el que usar más hilos no me garantiza un tiempo menor de ejecución, sino que éste se mantiene. Por otro lado, llega un punto en el que el algoritmo implementado no se pueda paralelizar más, y el desempeño queda en manos del algoritmo y no en la cantidad de hilos.

2. Cómo se comporta la solución usando tantos hilos de procesamiento como núcleos comparado con el resultado de usar el doble de éste?.

    a. Al usar el doble de núcleos de procesamiento se ve una reducción del tiempo a la mitad, un leve crecimiento en el uso de CPU del 0,7%, además de una disminución en el uso de memoria, ésto con respecto al uso de tantos hilos de procesamiento.

3. De acuerdo con lo anterior, si para este problema en lugar de 100 hilos en una sola CPU se pudiera usar 1 hilo en cada una de 100 máquinas hipotéticas, la ley de Amdahls se aplicaría mejor?. Si en lugar de esto se usaran c hilos en 100/c máquinas distribuidas (siendo c es el número de núcleos de dichas máquinas), se mejoraría?. Explique su respuesta.
    a. Tal vez habría un menor uso de recursos, pero como lo mencionaba anteriormente, llega un punto en el que el mejoramiento del desempeño en el algoritmo ya no dependerá de la cantidad de hilos o máquinas, sino de él mismo.