

# **Laboratorio 01 ARSW**

## **Juan David Ramírez Mendoza**

### **Parte I hilos Java**

- IV. Cambie el inicio con 'Start ()' por 'run ()'. ¿Cómo cambia la salida?, por qué?

Al ejecutar el método Start () Los Threads ejecutan sus tareas al mismo tiempo (se intercalan para ejecutar su proceso), por lo que la salida no tiene orden alguno, en cambio cuando se ejecuta el método run () los Threads ejecutan sus labores de manera secuencial, es decir, la salida tendrá el orden en que se les haya ejecutado con el método run ().

### **SALIDA CON START ()**

```
204
205
206
207
208
209
210
211
212
213
214
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
```

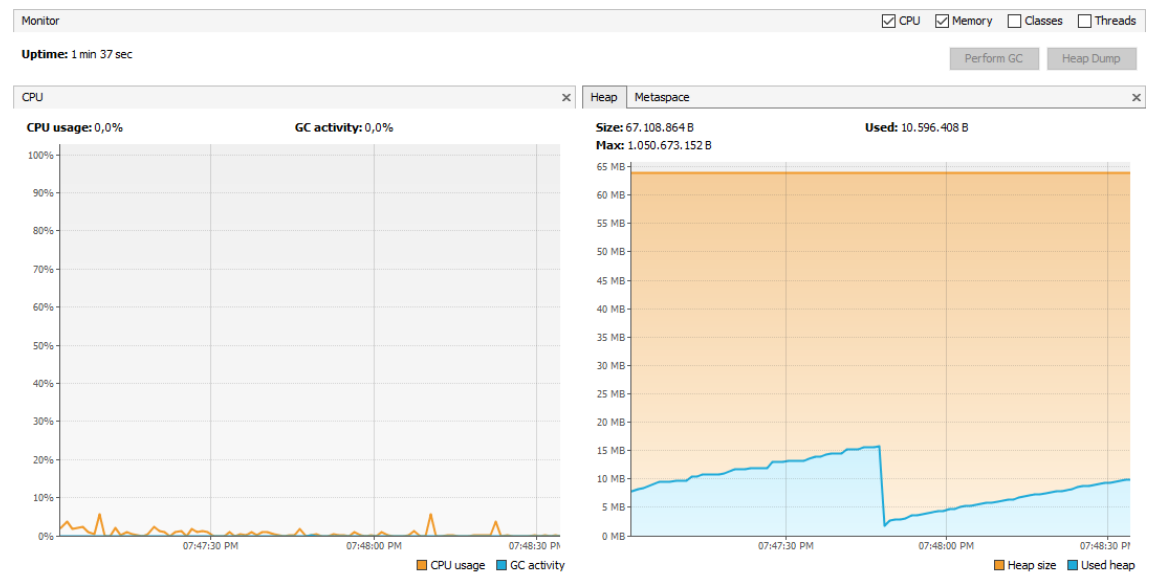
## SALIDA CON RUN ()

```
87
88
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
```

## Parte III Evaluación de Desempeño

### Pruebas:

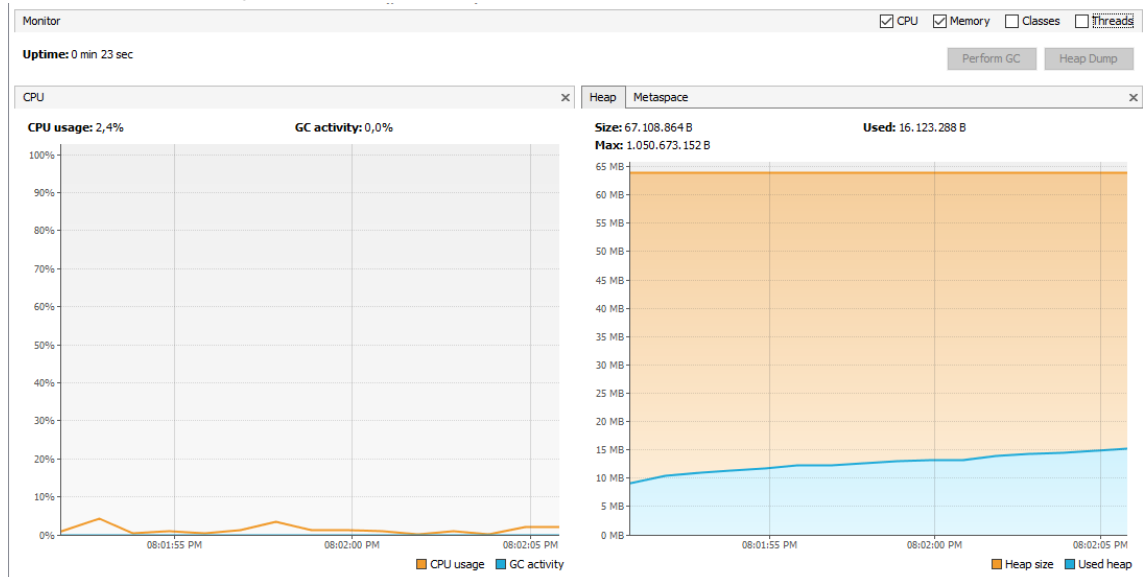
#### 1. 1 Hilo



El tiempo de ejecución según Java visualVM fue 1 minuto y 36 segundos. La CPU tuvo un máximo uso del 5.8% y el consumo de memoria fue de 10.596.408 B.

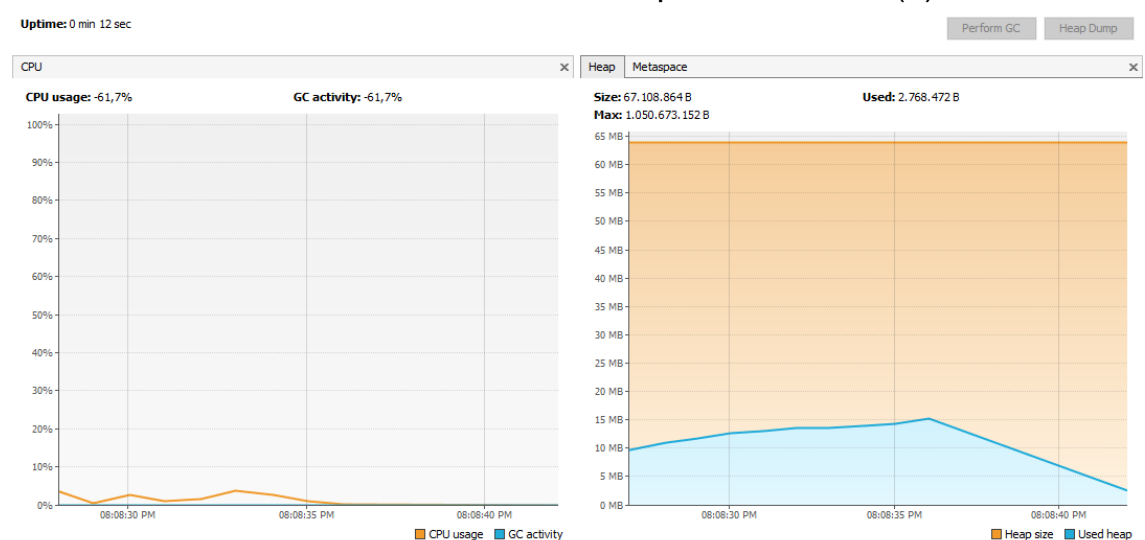
2. Tantos hilos como núcleos de procesamiento.

Para esta prueba se usó el método `availableProcessors()` de la clase `Runtime` (la CPU de la maquina en la que estoy trabajando tiene 4 núcleos)



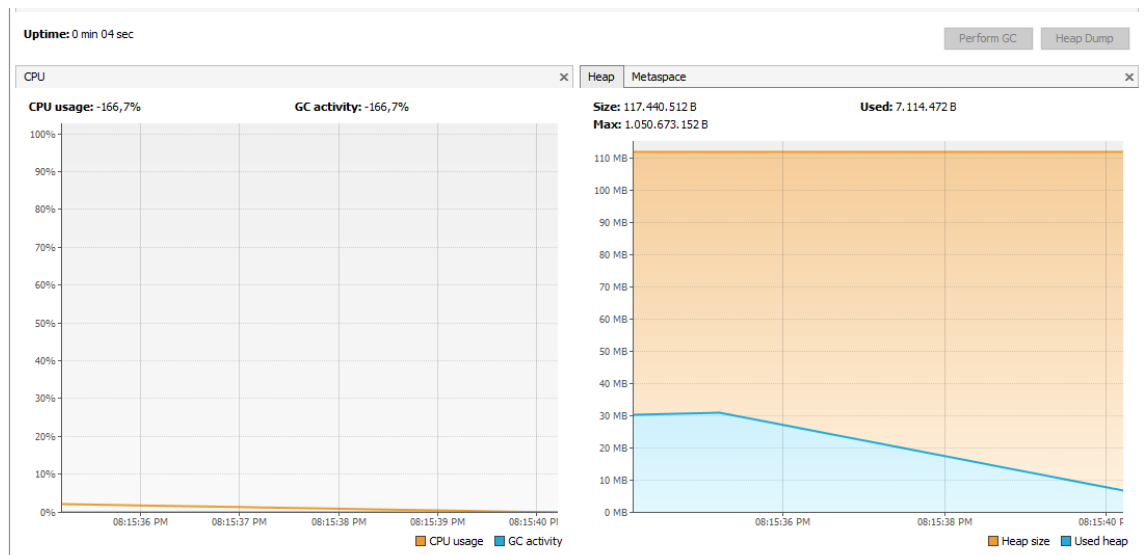
El tiempo de ejecución fue 23 segundos. La CPU tuvo un máximo consumo de 4.4%. Además, el consumo de memoria fue de 16.123.288 B.

3. Tantos hilos como el doble de núcleos de procesamiento (8)



El tiempo de ejecución fue de 12 segundos. La CPU tuvo un máximo consumo de 4.0% y el consumo de memoria fue de 2.768.472 B

4. 50 hilos.



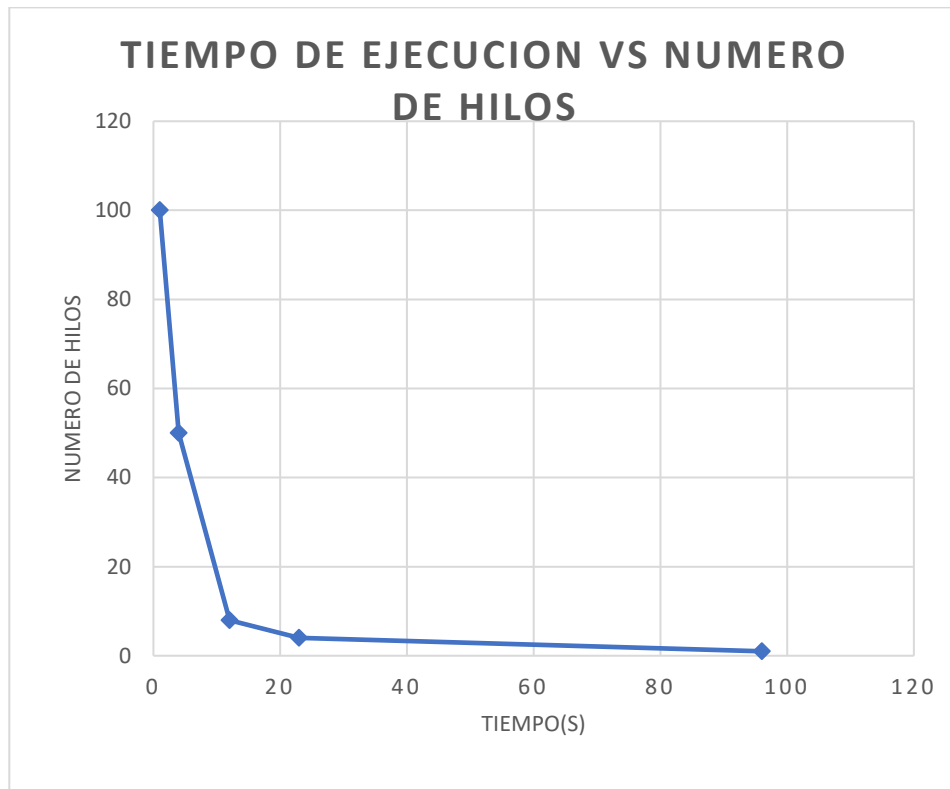
El tiempo de ejecución fue de aproximadamente 4 segundos. La CPU tuvo un máximo consumo de 2.2% y el consumo de memoria fue de 7.114.472 B.

## 5. 100 hilos



El tiempo de ejecución fue de aproximadamente 1 segundo. Como se puede ver no se pudo observar la grafica del consumo de CPU y se consumió en total 1.685.440 B de memoria.

## GRAFICA REALIZADA:



## Preguntas:

1. ¿Por qué el mejor desempeño no se logra con los 500 hilos?, cómo se compara este desempeño cuando se usan 200?  
Porque tal como se pudo observar en jvisualVM cada vez que se aumentan los hilos es verdad que se usa menos porcentaje de CPU, pero también es cierto que aumenta el uso de la memoria heap de la máquina, lo cual ocasiona que el tiempo empiece a aumentar a medida que se esta usando mucha mas memoria que al usar menos hilos. Cuando se usan 500 hilos el tiempo de respuesta es 1.449 s aproximadamente, al usar 200 hilos el tiempo es 1.500 s aproximadamente. Esto significa que si se siguiera aumentando la cantidad de hilos la gráfica sería una parábola ya que empezaría a aumentar el tiempo de respuesta.
2. ¿Cómo se comporta la solución usando tantos hilos de procesamiento como núcleos comparado con el resultado de usar el doble de éste?

Al usar el mismo numero de hilos que de núcleos se ejecutó en 23 segundos el proceso, al usar el doble de hilos se tardo 12 segundos.

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.

Si, la ley de Amdahls se aplicaría mejor ya que al usar más CPUS el máximo uso de la misma sería muy bajo en cada uno de los computadores lo cual sería más productivo. Por tal razón actualmente las maquinas cuentan con 2 CPUS (para simular lo anteriormente dicho).