

Parte I Hilos Java

- i. Cambie el inicio con 'start ()' por 'run ()'. ¿Cómo cambia la salida? ¿por qué?

R: Se imprime la respuesta de forma sincronizada, ósea de acuerdo al orden de código del programa. Porque al llamar al método run () no se están creando hilos o subprocesos nuevos, solo se está llamando a un método como cualquier otro.

Parte III Hilos Java

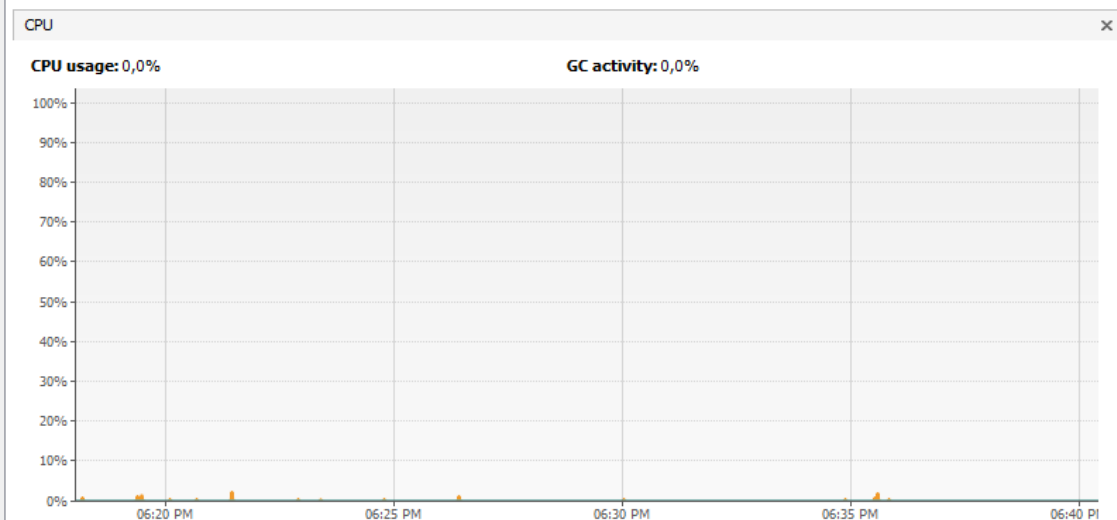
200.000 dígitos (hex) de PI.

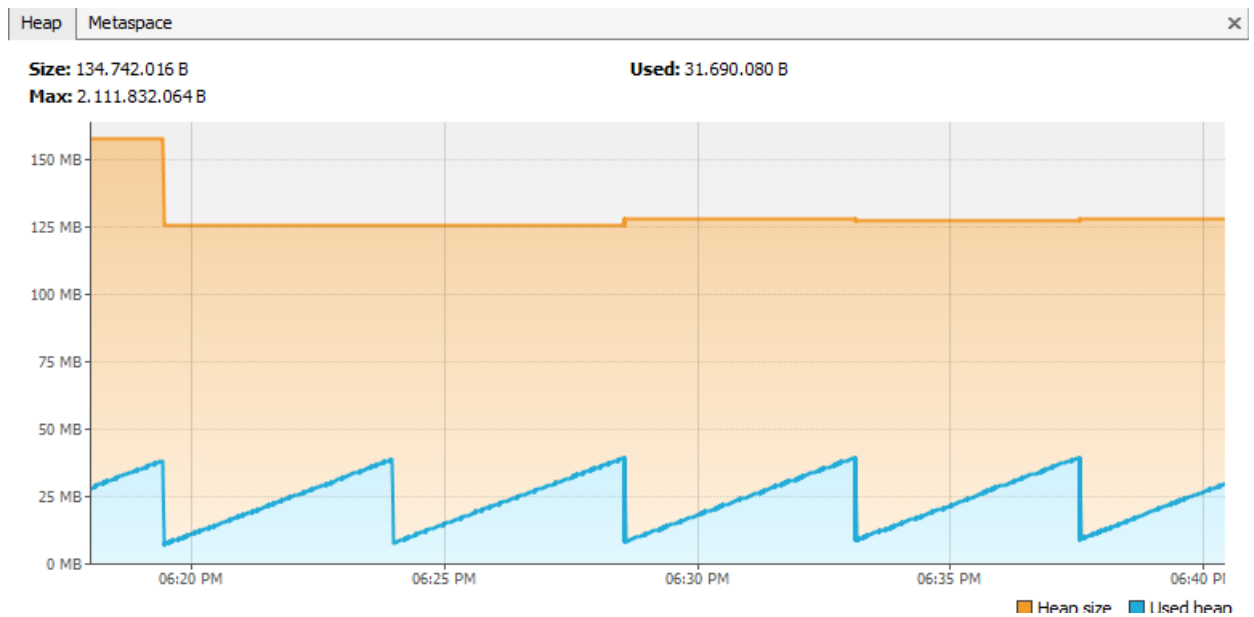
1.

1. Un solo hilo. **Tiempo de ejecución: 22:36.174s**

```
L 243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89452821E638D(
-----
BUILD SUCCESS
-----
Total time: 22:36.174s
Finished at: Mon Aug 14 18:40:27 COT 2017
Final Memory: 5M/121M
```

Uptime: 22 min 36 sec





2. Tantos hilos como núcleos de procesamiento (haga que el programa determine esto haciendo uso del [API Runtime](#)). En este Caso son 8. **Tiempo de ejecución: 6:10.054s**

```
243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89452821E63
```

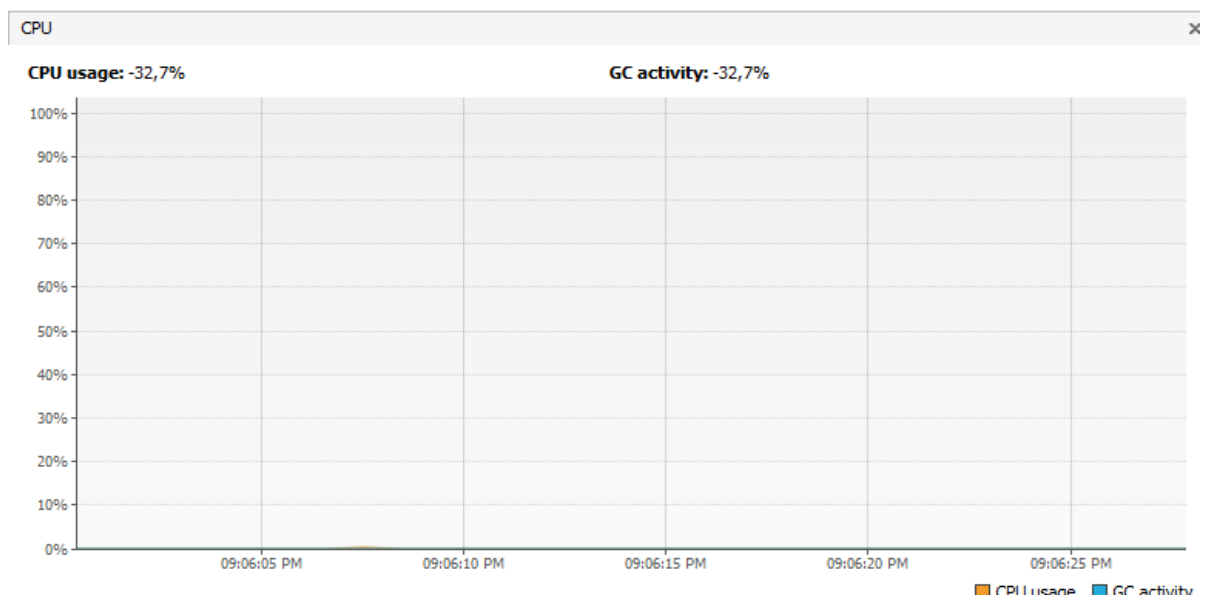
BUILD SUCCESS

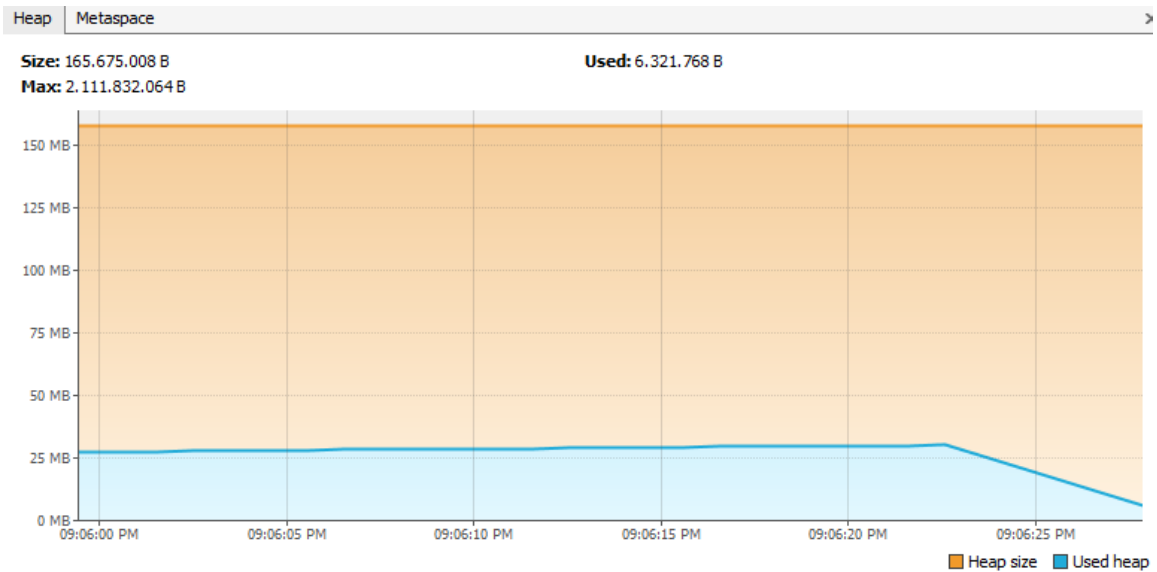
Total time: 6:10.054s

Finished at: Mon Aug 14 21:06:23 COT 2017

Final Memory: 5M/121M

Uptime: 6 min 10 sec





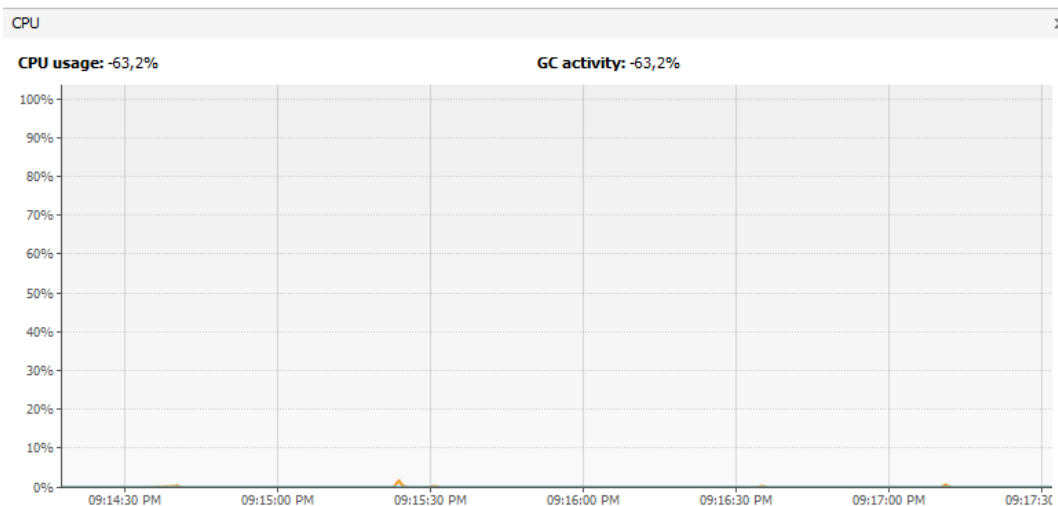
3. Tantos hilos como el doble de núcleos de procesamiento. En este caso son 16.
Tiempo de ejecución: 4:27.957s

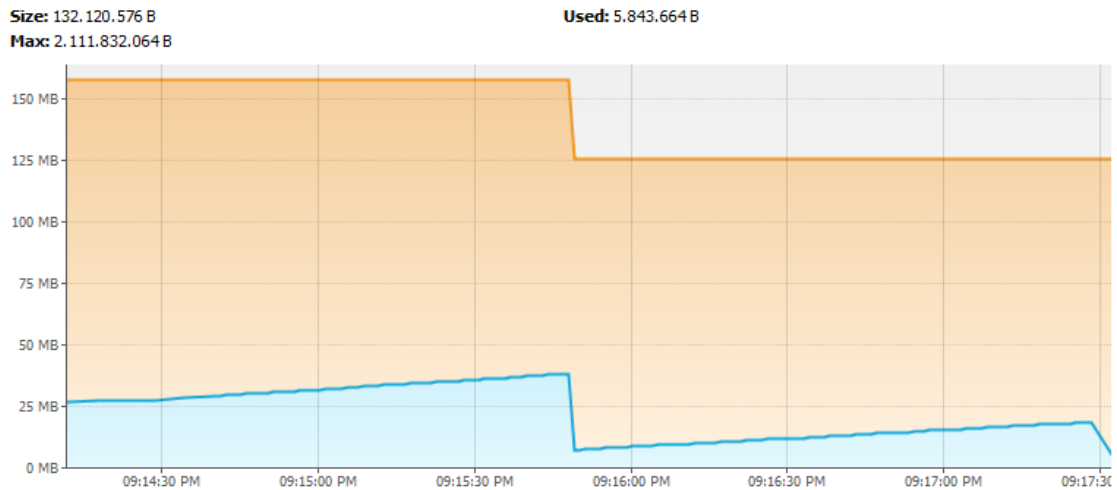
```

] --- exec-maven-plugin:1.2.1:exec (default-cli) @ PiDigits ---
· 243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89452821E63
-----
BUILD SUCCESS
-----
Total time: 4:27.957s
Finished at: Mon Aug 14 21:17:28 COT 2017
Final Memory: 5M/121M

```

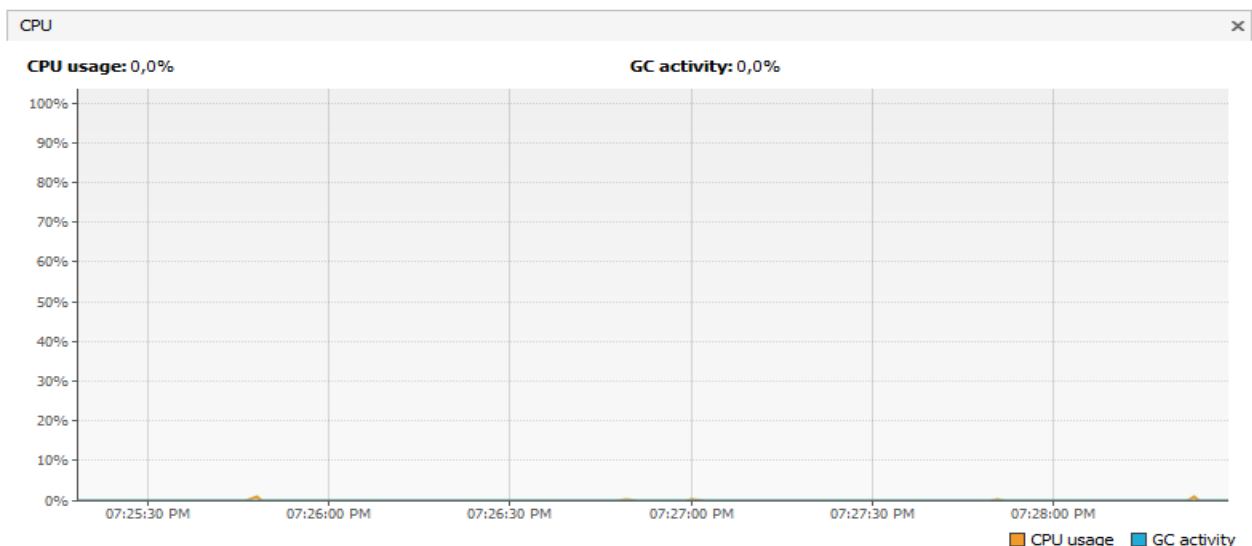
Uptime: 4 min 28 sec

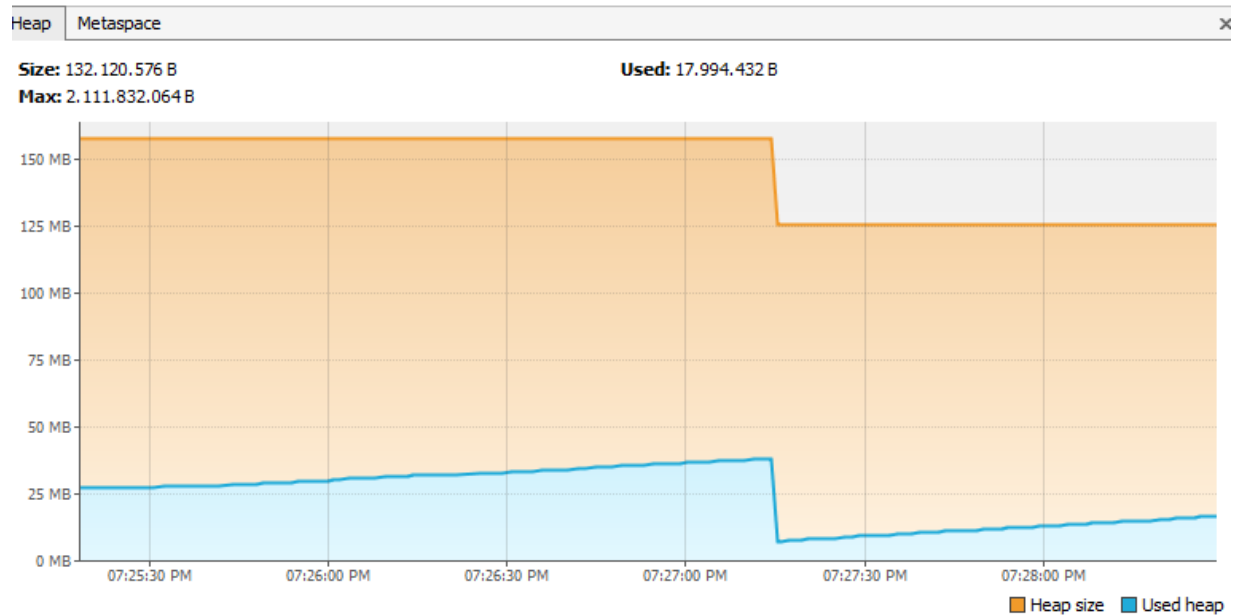




4. 200 hilos. Tiempo de ejecución: 3:55.420s

```
[ --- exec-maven-plugin:1.2.1:exec (default-cli) @ PiDigits ---  
- 243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89452821E638  
-----  
BUILD SUCCESS  
-----  
Total time: 3:55.420s  
Finished at: Mon Aug 14 19:28:29 COT 2017  
Final Memory: 5M/121M  
-----  
Uptime: 3 min 55 sec
```

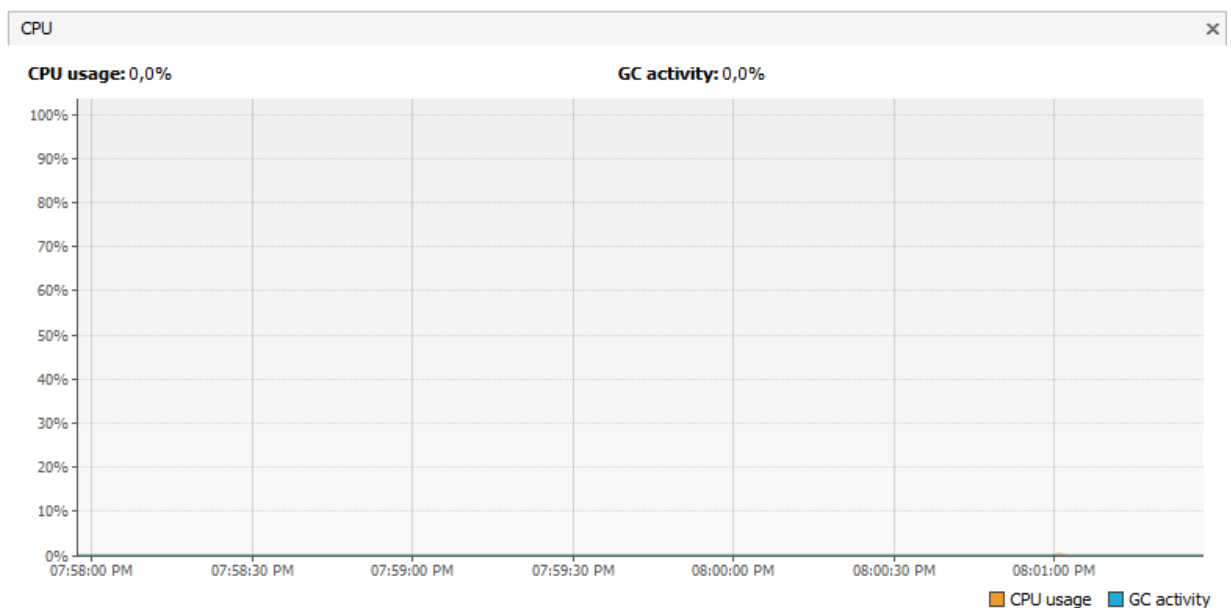


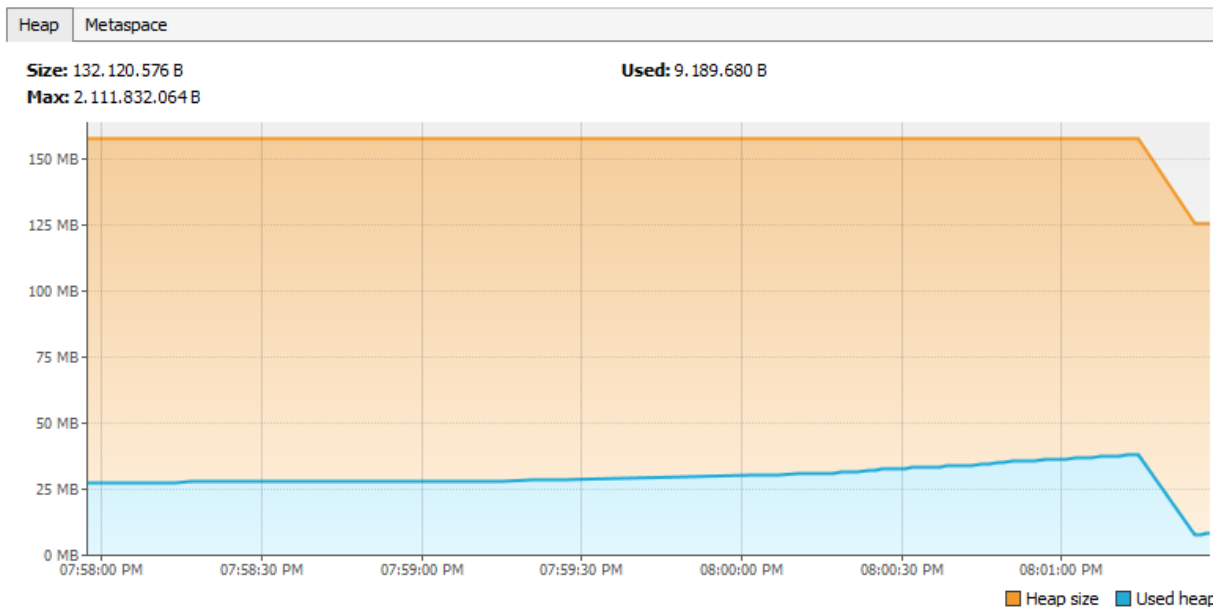


5. 500 hilos. Tiempo de ejecución: 3:55.229s

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ PiDigits ---  
243F6A8885A308D313198A2E03707344A4093822299F31D0082EFA98EC4E6C89452821E638  
-----  
BUILD SUCCESS  
-----  
Total time: 3:55.229s  
Finished at: Mon Aug 14 20:01:28 COT 2017  
Final Memory: 5M/121M  
-----
```

Uptime: 3 min 55 sec





2.

$$S(n) = \frac{1}{(1 - P) + \frac{P}{n}}$$

1. Según la [ley de Amdahls](#):

, 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?

R: Con 500 hilos no se logra el mejor desempeño porque ya se ha alcanzado un máximo mejoramiento de desempeño, lo cual no garantiza que al hacer más hilos se haga un cambio considerable en el desempeño. El desempeño con 500 (T: 3:55.229s) hilos comparado con el desempeño con 200 (T: 3:55.420s) hilos es imperceptible ya que ambas ejecuciones duran casi el mismo tiempo, por lo tanto se puede decir que no se logra un mejor desempeño con la ejecución de los 500 hilos.

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

R: El desempeño usando el doble de núcleos de procesamiento mejora considerablemente en aproximadamente 2 minutos.

3. De acuerdo con lo anterior, si para este problema en lugar de 500 hilos en una sola CPU se pudiera usar 1 hilo en cada una de 500 máquinas hipotéticas, ¿la ley de Amdahls se aplicaría mejor? ¿Si en lugar de esto se usaran c hilos en $500/c$ máquinas distribuidas (siendo c es el número de núcleos de dichas máquinas), se mejoraría? Explique su respuesta.

R: No, porque se desperdiciaría tiempo de procesamiento utilizando un único hilo por máquina, lo que se requiere es usar su máxima capacidad de núcleos por proceso de las máquinas. Sí, porque cada máquina tendría un c hilos ejecutándose al máximo de su capacidad por núcleos del proceso de la máquina.