

Programación Concurrente

UCM – Facultad de Informática
Grado en Ingeniería Informática
2012-2013
Curso 3º A

Juan Luis Álvarez Herradón
Ángel Luis Ortiz Folgado



UNIVERSIDAD COMPLUTENSE
MADRID

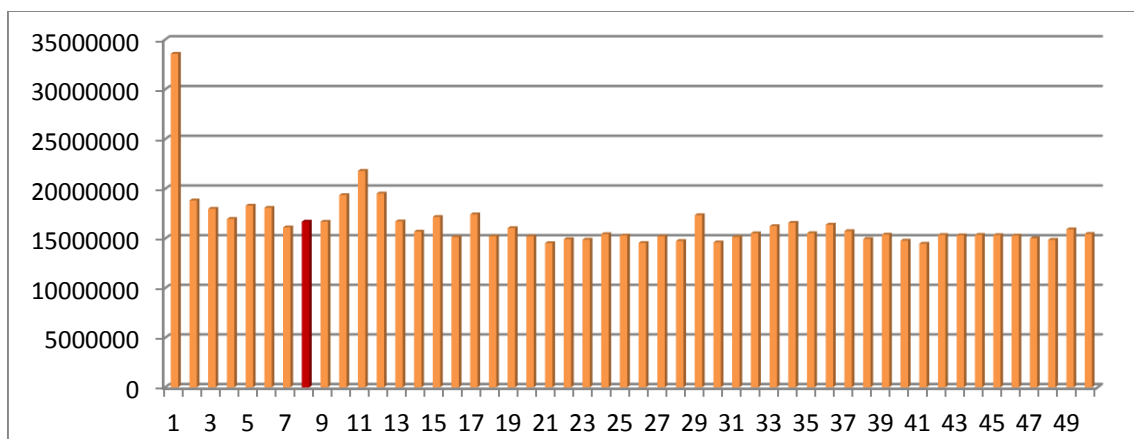
[MEMORIA DE LA PRÁCTICA 5]

PREGUNTAS

EJERCICIO 1

El resultado tras la ejecución ha sido el siguiente (en rojo, el tiempo para 8 hilos con 8 cores):

| | | |
|-------------------------------------|----|----------|
| Numero de procesadores disponibles: | 24 | 15446818 |
| 8 | 25 | 15298470 |
| Numero de hilos | 26 | 14552067 |
| | 27 | 15214033 |
| 1 | 28 | 14754529 |
| 2 | 29 | 17357144 |
| 3 | 30 | 14605248 |
| 4 | 31 | 15150123 |
| 5 | 32 | 15531255 |
| 6 | 33 | 16250135 |
| 7 | 34 | 16585550 |
| 8 | 35 | 15543384 |
| 9 | 36 | 16396616 |
| 10 | 37 | 15741181 |
| 11 | 38 | 14962123 |
| 12 | 39 | 15390371 |
| 13 | 40 | 14788584 |
| 14 | 41 | 14477893 |
| 15 | 42 | 15362847 |
| 16 | 43 | 15323195 |
| 17 | 44 | 15366579 |
| 18 | 45 | 15352584 |
| 19 | 46 | 15294272 |
| 20 | 47 | 15017169 |
| 21 | 48 | 14868822 |
| 22 | 49 | 15928249 |
| 23 | 50 | 15470143 |



EJERCICIO 2

Al contrario que en el ejercicio anterior, aquí la clase *TareaLarga* implementa *Callable*, para que desde el *CompletionService* se pueda usar *take()*, de forma que espere si no ha terminado ninguna tarea.

Un ejemplo de salida es el siguiente:

```
La tarea 0: pool-1-thread-10
La tarea 1: pool-1-thread-7
La tarea 2: pool-1-thread-6
La tarea 3: pool-1-thread-1
La tarea 4: pool-1-thread-8
La tarea 5: pool-1-thread-9
La tarea 6: pool-1-thread-3
La tarea 7: pool-1-thread-5
La tarea 8: pool-1-thread-4
La tarea 9: pool-1-thread-2
```