

MEMORIA PRÁCTICA 5

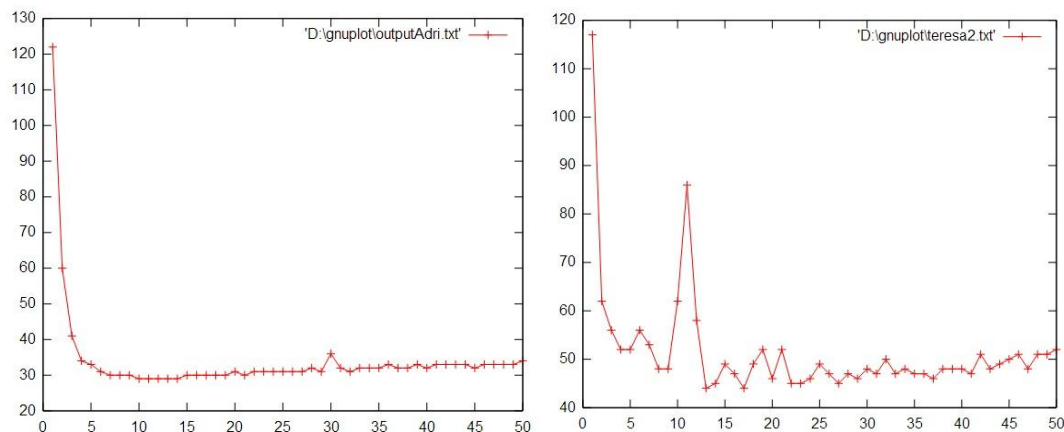
Teresa Rodríguez Ferreira

Adrián Rabadán Jurado

Ejercicio 1:

Para este apartado hemos creado la clase TareaDeCalculo (que implementa Runnable), lo único que hace es contar hasta un número que le pasamos en la constructora. A continuación hemos modificado la clase EjemploExecutorService como se pedía, utilizando un ThreadPool para crear los hilos, y presentando instancias de la clase TareaDeCalculo al ExecutorService. El programa mide el tiempo que tarda en contar hasta el número en función de la cantidad de hilos que se hayan creado en el pool. Se observa que la eficiencia aumenta a medida que el número de hilos crece, hasta que sobrepasa el umbral del procesador que lo ejecute, pues tiene que gestionar más hilos de los que puede ejecutar paralelamente. Hemos utilizado shutdown, awaitTermination y shutdownNow para cerrar el pool si tiene demasiado coste en tiempo.

Ejemplo de ejecuciones en ordenadores distintos:



Ejercicio 2:

Ahora creamos la clase TareaLarga (que implementa Callable), que simplemente espera durante un tiempo largo aleatorio, y después devuelve el nombre del hilo (asignado por defecto). En la clase EjemploCompletionService empleamos una lista de TareasLargas y se las presentamos al CompletionService para que las ejecute.

Ejercicio 3:

En este ejercicio hemos modificado las clases para que utilicen ForkJoinPool (java7) en vez de ThreadPool. Hemos modificado lo que se pedía en las pistas y simplemente hemos tenido que cambiar algunos métodos para que el funcionamiento sea el mismo que en la versión anterior. Hemos reducido el número de enlaces para poder comprobar mejor su funcionamiento.