

Iteración 1:

```
useer@useer-VirtualBox:~/Documents/GitHub/corto2_paralela$ ./iteracion 40
El 40-ésimo término de Fibonacci es: 102334155
Tiempo de ejecución: 1.052093 segundos
```

Iteración 2:

```
useer@useer-VirtualBox:~/Documents/GitHub/corto2_paralela$ ./iteracion2 40
El 40-ésimo término de Fibonacci es: 102334155
Tiempo de ejecución: 8.781126 segundos
```

Iteración 3:

```
useer@useer-VirtualBox:~/Documents/GitHub/corto2_paralela$ ./iteracion3 40
El 40-ésimo término de Fibonacci es: 102334155
Tiempo de ejecución: 1.209541 segundos
useer@useer-VirtualBox:~/Documents/GitHub/corto2_paralela$
```

Pregunta 1:

Lo que está pasando internamente es que al introducir la paralelización de tareas en *fib_recursive_omp()* se está dando una sobrecarga por la creación y sincronización de tareas. Es por ello que se ve que el tiempo de ejecución empeora de casi un segundo a 8 segundos. En la iteración 3 al utilizar el threshold, se está solucionando esta sobrecarga. El threshold lo que hace es determinar el momento en el que es conveniente crear una tarea adicional o si es mejor seguir secuencial.

Pregunta 2:

Como se mencionó antes, el threshold es este valor "umbral" que se utiliza para determinar cuándo es conveniente crear una tarea adicional o simplemente realizar el cálculo de forma secuencial. Al establecer un threshold, se evita la creación de tareas para cálculos pequeños, lo que reduce la sobrecarga y mejora el rendimiento. Si n es menor o igual al valor de threshold, entonces el cálculo del término Fibonacci se hace de manera secuencial utilizando la función *fib_recursive(n)*. Por otro lado, si n es mayor que el valor de threshold, se crea un conjunto de tareas paralelas para calcular los términos $(n-1)$ y $(n-2)$ y luego se espera a que estas tareas finalicen utilizando `#pragma omp taskwait`.

Tabla:

Fibonacci de 40

Iteración 2

	Tiempo
1	9.532612
2	9.349770
3	9.207852
4	9.345379
5	9.355435
Promedio	9.3582096

Tiempo Iteración 2	Tiempo Iteración 3	Hilos	Threshold	Speedup
1.356160	9.3582096	1	10	6.900520293
9.839358	9.3582096	10	1	0.951099614
2.565853	9.3582096	5	5	3.647211902
1.359458	9.3582096	5	10	6.883779859
1.372284	9.3582096	4	10	6.819440874
1.231049	9.3582096	5	15	7.601817312
1.169524	9.3582096	5	30	8.001725146
1.166867	9.3582096	5	100	8.019945375

Como se puede ver en la tabla anterior, un threshold demasiado bajo resulta en una sobrecarga de creación y manejo de tareas paralelas, lo que hace que la versión paralela sea más lenta que la versión secuencial debido a la cantidad de tareas pequeñas que se están creando. Sin embargo, un threshold demasiado alto puede significar que se están calculando tareas recursivas de manera secuencial incluso cuando podrían beneficiarse de la paralelización. En este caso al aumentar el threshold, el tiempo de ejecución mejora. Sin embargo, existe un punto en el que este beneficio se comienza a estabilizar, ya que agregar tareas paralelas adicionales genera sobrecarga debido a la administración de esas tareas.