

# Práctica 2

Manuel Pérez Ruiz

27 de octubre de 2022

## 1. Ejercicio 3

Para la realización de este ejercicio tenemos, como se nos pide, dos archivos, **escalaVector.java** y **EscalaVPar.java**.

### 1.1. EscalaVector.java

En el caso de **escalaVector.java**, he creado la clase `escalaVector`, en la que he declarado los atributos *vector*, *escala* y *tamanno*, en los que almacenaré el vector que vamos a escalar, el factor por el que vamos a escalar los elementos del vector y el tamaño máximo del vector respectivamente.

En el main, una vez hemos inicializado un vector de enteros y le hemos asignado un valor aleatorio a cada posición del vector, pasamos a construir un objeto de la clase `escalaVector` y llamamos al método `run`, que se encarga de escalar el vector.

Como podemos ver en las figuras 1 y 2, no tenemos ningún pico reseñable ni con un factor de escala 1, ni con un factor de escala 5, salvo en los núcleos 1, 3 y 5, que suben un poco más que el resto de núcleos.

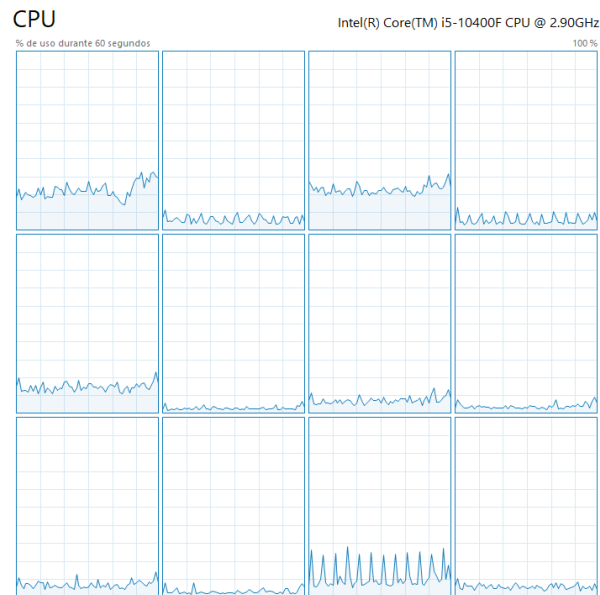


Figura 1: Factor de escala 1.

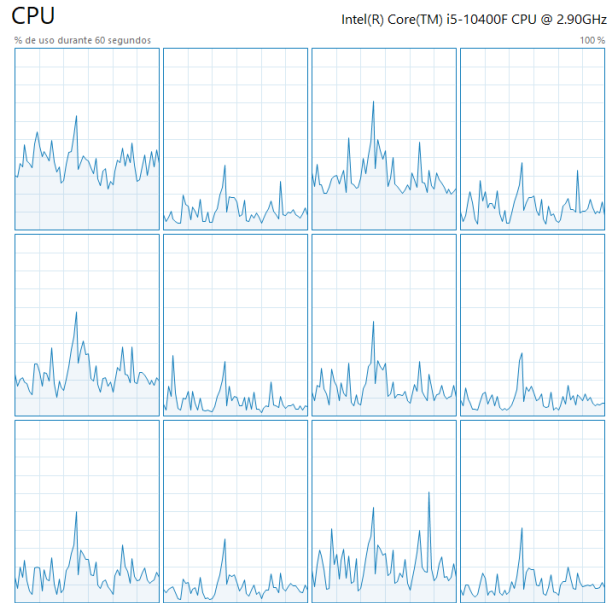


Figura 2: Factor de escala 5.

## 1.2. EscalaVPar.java

En este caso, he creado la clase `escalaVPar`, en la que, a diferencia del caso anterior, solo he declarado el atributo *tamanno*, en el que almacenamos el tamaño máximo del vector.

También he creado la clase *hebra*, que hereda de *Thread*. Aquí declaramos los atributos *vector* (en el que vamos a almacenar el vector), *escala* (el factor de escala que vamos a usar), *inicio* (la posición inicial del vector que va a usar cada hebra para escalar) y *fin* (la posición final del vector que va a escalar la hebra en cuestión). Por último, he declarado el método `run`, en el que escalamos los elementos del vector.

En el `main` (que está en la clase `escalaVPar`), una vez hemos inicializado un vector de enteros y le hemos asignado un valor aleatorio a cada posición del vector, pasamos a construir 4 hebras, las iniciamos y las ejecutamos.

Como podemos observar en las figuras 3 y 4, no hay una diferencia significativa, teniendo en cuenta que el factor de escala es de 2 y de 15 respectivamente. Vemos, al igual que en el apartado anterior, los núcleos 1, 3 y 5 son los que más "trabajan".

## 1.3. Comparación final

Comparando las gráficas de la versión secuencial y la paralela, no podemos observar una diferencia en lo que a los picos se refiere, aunque sí podemos ver cómo el rendimiento de la CPU en las figuras 1 y 2 (la versión secuencial) es menor que en las figuras 3 y 4 (la versión paralela), es decir, que a la CPU le cuesta más "trabajo" llevar a cabo la versión secuencial.

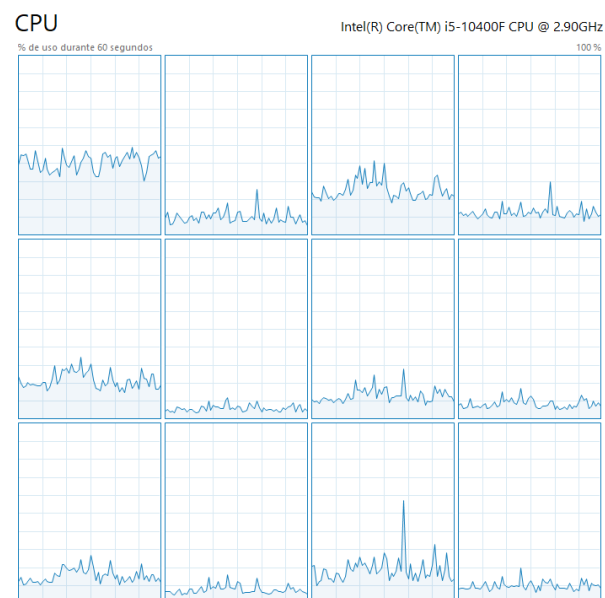


Figura 3: Factor de escala 2.

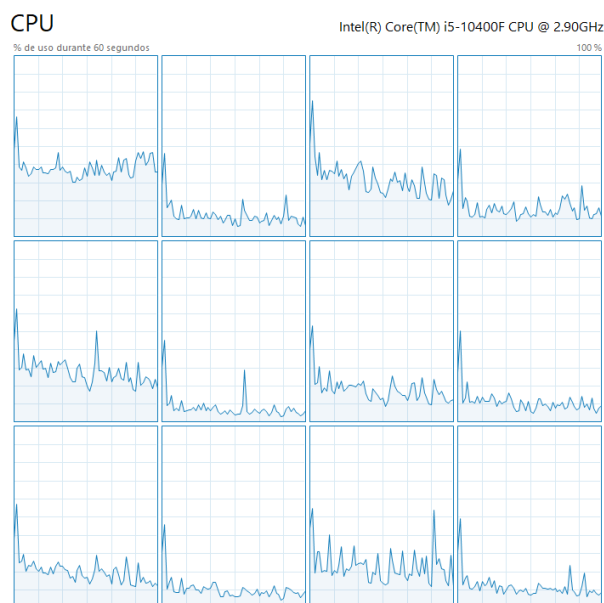


Figura 4: Factor de escala 15.