

# CUDA WORKSHOP 2018

---

## PRÁCTICAS

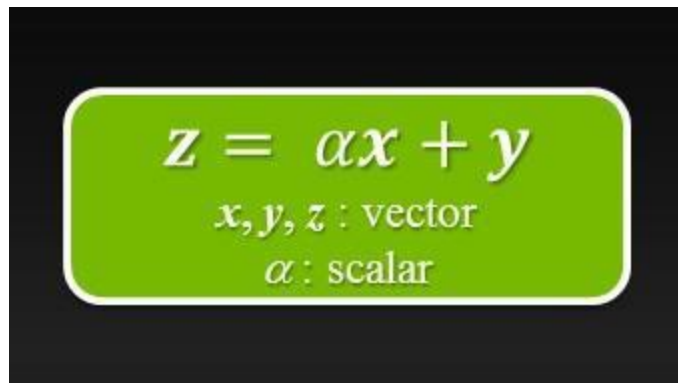
Sergio Orts Escolano (sorts@ua.es)

Albert García García (agarcia@dtic.ua.es)

José García Rodríguez (jgarcia@dtic.ua.es)

### PRÁCTICA 6: ZAXPY USANDO THRUST

En este laboratorio se propone implementar una operación paralela simple como puede ser la operación SAXPY (o ZAXPY) explicada en figura 1.

El diagrama muestra la ecuación  $z = \alpha x + y$  en un recuadro verde con esquinas redondeadas. Debajo de la ecuación, se especifica que  $x, y, z$  son vectores y  $\alpha$  es un escalar.
$$z = \alpha x + y$$

$x, y, z$  : vector  
 $\alpha$  : scalar

figura 1. operación zaxpy

**Paso 1:** Para empezar se propone implementar la operación simple de zaxpy. Para simplificar realizaremos la operación  $Y[i] = a * X[i] + Y[i]$ .

En la función main debes crear inicializados los correspondientes device\_vectors que contendrán los vectores X e Y, y traerlos de vuelta de nuevo volcándolos en los vectores X e Y (el array Y tendrá la solución).

Siguiendo el esquema propuesto en la función `zaxpy_slow`;

1. Inicializamos un `device_vector` `temp` relleno con los valores del escalar `A`
2. Multiplicamos el vector temporal y el vector `X`, consiguiendo `AX`
3. Por último sumamos el vector temporal (ahora con `AX`) con el vector `Y`

**Paso 2:** Una vez completado el `saxpy_slow`, se pide implementar el `saxpy_fast` que hará uso de un functor para realizar la operación en un solo paso. Podéis guiarnos por la teoría y en la documentación de la librería Thrust (<https://code.google.com/p/thrust/>).

**Paso 3:** Una vez vista la utilización de los functors, ¿qué cambios tendrías que realizar para hacer la operación  $Y[i] = a * X[i] + b * Y[i] + X[i] * Y[i]$ ?

**Paso 4:** ¿Qué tipo de operaciones y estructuras de Thrust se necesitarían para realizar una operación con más de dos vectores, por ejemplo,  $Z[i] = R[i] * S[i] * T[i]$ ? Prueba a implementar una transformada como la propuesta.