# **Fundamentals of Accelerated Computing with OpenACC**

# links:

<https://www.openacc.org/resources>

<https://www.openacc.org/community#slack>

<https://docs.nvidia.com/nsight-systems/>

<https://developer.nvidia.com/nsight-systems>

<https://docs.nvidia.com/cuda/profiler-users-guide/index.html#nvtx>

# cmd

## compile the code

!nvc -fast -o laplace -Mprof=ccff -I/opt/nvidia/hpc\_sdk/Linux\_x86\_64/21.3/cuda/11.2/include jacobi.c laplace2d.c && echo "Compilation Successful!" && ./laplace

## compile the code again with the -Minfo=opt flag, which instructs the compiler to print additional information how it optimized the code

!nvc -fast -Minfo=opt -I/opt/nvidia/hpc\_sdk/Linux\_x86\_64/21.3/cuda/11.2/include -o laplace jacobi.c laplace2d.c

## Run Our Parallel Code on Multicore CPU

!nvc -fast -ta=multicore -Minfo=accel -I/opt/nvidia/hpc\_sdk/Linux\_x86\_64/21.3/cuda/11.2/include -o laplace\_parallel ./solutions/parallel/jacobi.c ./solutions/parallel/laplace2d.c && ./laplace\_parallel

## Run Our Parallel Code on GPU

!nvc -fast -ta=tesla:managed -Minfo=accel -o laplace\_managed jacobi.c laplace2d.c && ./laplace\_managed

# Flags

**-Minfo** : This flag will give us feedback from the compiler about code optimizations and restrictions.

**-Minfo=accel** will only give us feedback regarding our OpenACC parallelizations/optimizations.

**-Minfo=all** will give us all possible feedback, including our parallelizaiton/optimizations, sequential code optimizations, and sequential code restrictions.

**-ta** : This flag allows us to compile our code for a specific target parallel hardware. Without this flag, the code will be compiled for sequential execution.

**-ta=multicore** will allow us to compiler our code for a multicore CPU.

# Introduction to Parallel Programming with OpenACC

<https://www.youtube.com/watch?v=PxmvTsrCTZg&list=PLx_s9Cz7_T429SF7gBGJ51iiZoEWYVvkq>