# Session 6: Advanced OpenMP Offload

*Christian Terboven, Paul Kapinos*
*IT Center, RWTH Aachen University*
*Seffenter Weg 23, 52074 Aachen, Germany*
*{terboven, kapinos}@itc.rwth-aachen.de*

*Michael Klemm*
*AMD; OpenMP ARB*
*michael.klemm@openmp.org*

*Yun (Helen) He*
*NERSC, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*
*yhe@lbl.gov*

## Abstract

This document guides you through the exercises. Please follow the instructions given during the training session.

The prepared "Makefile" builds the executables jacobi.gpu and jacobi.sol.gpu using Nvidia compilers on Perlmutter GPU.

The provided "run.sh" is a batch script that you can use to submit a batch job to run on a Perlmutter GPU compute node. Submit the job via command "sbatch run.sh", and check the output file after it is run, pay attention to total run time.

You can also run an interactive batch job via "salloc" to get on a compute node. For example:

```
% salloc -N 1 -q interactive -C gpu -G 1 -c 32 -t 30:00 -A ntrain5 <will land on a compute node>
% module load PrgEnv-nvidia
% make

% OMP_NUM_THREADS=1 ./jacobi.gpu
% OMP_NUM_THREADS=1 ./jacobi_sol.gpu
```

Please refer to https://github.com/NERSC/openmp-series-2024/blob/main/Session-5-OpenMP-Offload/Using-OpenMP-Compilers-on-Perlmutter-GPUs-Sept2024.pdf for more details on Using various OpenMP compilers on Perlmutter GPUs and running jobs.

## 1. Jacobi on GPU

In this exercise, you will port a Jacobi solver to OpenMP to run on GPUs. This Jacobi example solves a finite difference discretization (5-point-stencil) of the Laplace equation (2D). Please find more info on Jacobi in Session 5 slides deck.

Continue Session 5 tasks 0-3. New task 4 for Session 6.

■ Task 0: You might want to acquire reference measurements on the host (i.e., on CPU, without/ GPU)...

■ Task 1: Get it to the GPU: Parallelize only the one most compute-intensive loop. Besides using Nvidia compiler, also try to compile the codes using LLVM/Clang compiler, CCE compiler, and GCC compiler.

■ Task 2: Improve the data management and the amount of parallelism on the GPU

■ Task 3: Optimize that scheduling of iterations for the GPU

Understand the performance of the host and the GPU

■ Task 4: Make the code as fast as you can :-).

Use sample codes in exercises/<C,Fortran>/Jacobi2 for hints

Future Tasks: use multiple GPUs, use the host and a GPU, ...