Hands-On Session 1

Dominik Zobel, Jan Frederik Engels

 |

DKRZ

Goal of this hands-on

Run a small program on GPU

i.e.

- compile code with OpenACC directives for offloading to a device
- use an appropriate SLURM runscript to submit the job on the GPU partition

Everybody should have successfully done this at the end of this hands-on!

Example program

Simple matmul implementation similar to the following

```
 1 program matmul_test
 2    ! ... Declarations
 3
 4    call cpu_time(start_time)           ! Time from after initialization
 5    do it = 1, niter
 6       !$acc parallel loop
 7       do j = 1, nel
 8          do i = 1, nel
 9             do k = 1, nel
10                mat_out(i,j) = mat_out(i,j) + mat_a(i,k) * mat_b(k,j)
11             end do
12          end do
13       end do
14       !$acc end parallel loop
15    end do
16    call cpu_time(end_time)             ! until prcessing is done
17 end program matmul_test
```

See also matmul.f90 file

Compilation on Levante

- Load the NVHPC package

```
1 module load nvhpc/22.5-gcc-11.2.0
```

- Compile it

```
1 nvfortran -acc=verystrict -gpu=cc80 -Minfo matmul.f90 -o matmul_nv
```

Note: *Compiling can be done on a login node or on the `interactive` partition. Specifically, it does not have to be on a GPU node.*

Run script

- SLURM script to submit the job

```
#!/bin/bash
#SBATCH --account=<account-number> # FIXME: Insert your group account h
#SBATCH --partition=gpu
#SBATCH --nodes=1
#SBATCH --gpus=1
#SBATCH --mem=20G
#SBATCH --time=00:04:00

./matmul_nv

exit 0
```

Tasks (1/2)

1. Login on Levante, get the matmul.f90 file
2. Load the NVHPC module and compile the code
3. Create the run script and submit the job

If any error occurs and you don't understand why → ask us for help!

Tasks (2/2)

Note: Optional tasks which will be revisited in the next hands-on

Investigate the following effects on the `matmul_naive()` subroutine:

4. What happens if `if(acc)` is removed from `!$acc parallel loop if(acc)`?
5. What happens if you replace the parallel statements with kernel statements? That is, `!$acc parallel loop if(acc)` and its end directive with `!$acc kernels if(acc)` and its end directive?
6. What happens if `!$acc parallel loop if(acc)` and its end directive is moved one level up, i.e around the the `do` loop with `it = 1, iter`?