Tutorial: BwUniCluster 3.0/HoreKa
Modules

In this tutorial we will learn about modules and how to utilize them on the BwUniCluster or HoreKa.

1. What are modules?

Modules are means by which the user can access and use the software installed on the BwUniCluster or HoreKa such as widely used compilers and software. In order to use a specific programm or software (e.g. OpenFOAM) the user has to load the corresponding module. It is an excellent concept that allows the users to immediately load and use any version of OpenFOAM that is currently installed (currently more than 10 versions are supported). Users do not need to do any installation!

2. How to find and load a specific module?

You can view the list of all availible modules using the following command:

```
$>   module av
```

or

```
$>   module spider openfoam
```

The module can be then loaded by using the following command:

```
$>   module load [name of the module]
```

For example:

```
$>   module load cae/openfoam/v2112
```

The computer then generates the following response:

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Don't forget to initialize OpenFOAM with:

     source $FOAM_INIT

or simply:

     foamInit
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

and in a second step the user has to use one of the above two commands to set up the environment varaibles for OpenFOAM. Loading the module alone is not enough.

Together with the OpenFOAM v2112 also the corresponding compilers and MPI packages (with which the OpenFOAM was compiled) are automatically loaded:

```
$> module list
returns:
Currently Loaded Modules (could differ depending on the supercomputer):
1) compiler/gnu/13   2) mpi/openmpi/5.0   3) cae/openfoam/v2112
```

After initializing OpenFOAM we can use it like we would on a personal computer. However, we need to remember that a module initialized this way can only be used in the current terminal window, i.e. opening a second terminal does not mean that the modules loaded on the first terminal will be availible in the second one!

Now, we are able to prepare a new run – make the grid, set up specific initial and boundary conditions, prepare the parallelisation etc. This preparation we usually carry out on the login-node – exactly as we would do it on our personal desktop or laptop.

Care needs to be taken when we submit the job. This job will be executed on a different part of the supercomputer (the so called run-nodes). The run-nodes also have to load the module of OpenFOAM before we can use them. Here, there is a difference: we do not do it interactively but instead, we put the commands in a script file that is (later on) used to submit the job. The commands of interest are given in blue, the rest of the file will be explained later. They should precede the call of the solver (in this case pimpleFoam).

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#!/bin/bash

#SBATCH --partition dev_cpu // if you are on bwUniCluster          //  dev_cpuonly (is you are on HoreKa)
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:30:00
#SBATCH --mem=8000mb
#SBATCH --job-name=MyFirstParallelJob

module purge
module load  cae/openfoam/v2112
source $FOAM_INIT

pimpleFoam
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

3. How to remove installed modules (in order to load others)?

To see what modules you have currently loaded (some of them are loaded automatically when you login) use the command:

```
$> module list
```

In order to remove the already loaded modules (all of them) we can use the following command:

```
$> module purge
```

To remove only one module, e.g. the compiler, we use:

```
$> module unload compiler/gnu/13
```