User applications and libraries

Modulefiles

On Isabella cluster we use Modulefiles tools to bring users support for multiple versions of the same software and applications. For each version of available software we made a module that defines environment variables that need to be activated for the selected software to work. The main command to use Modulefiles is **module.** All changes to environmental variables are made only on the active session, or the started job. For every new job module activation is needed.

To see all available modules use command:

```
module avail
```

Or for a specific software, e.g. mpi:

```
module avail mpi
```

To activate a module, to set the environmental variables:

```
module load module_name
```

To change active module version:

```
module switch module_name new_version
```

Example of MPI environment change:

```
# module load mpi/mvapich2-intel-2.2-x86_64
# which mpicc
/usr/lib64/mvapich2-intel-2.2/bin/mpicc
# module switch mpi mpi/openmpi3-intel-x86_64
# which mpicc
/usr/lib64/openmpi3-intel/bin/mpicc
```

To see all loaded modules, either through load command or by other modules:

```
module list
```

Unloading a module, or removing environmental variables of that module:

```
module unload module_name
```

Unloading all modules:

```
module purge
```

To see all changes to environmental variables a specific module will do:

```
module show module_name
```

Compilers

Intel

The table shows all available Intel compiler versions, and their modules. We advice you to use the latest version - **Intel 2020**.

Version	Module
Intel 2019	intel/2019
Intel 2018	intel/2018
Intel 2017	intel/2017

For compiling use:

- icc - C compiler
- ifort - Fortran compiler

Intel MKL (Math Kernel Library) is available at $MKLROOT. Documentation on MKL: Developer Guide.

Parameters for compiling parallel applications in Fortran:

```
export FFLAGS="-i8 -I${MKLROOT}/include/intel64/ilp64 -I${MKLROOT}/include"
export FCFLAGS=$FFLAGS
export LIBS="${MKLROOT}/lib/intel64/libmkl_blas95_ilp64.a ${MKLROOT}/lib/intel64/libmkl_lapack95_ilp64.a -
L${MKLROOT}/lib/intel64 -lmkl_scalapack_ilp64 -lmkl_intel_ilp64 -lmkl_sequential -lmkl_core -
lmkl_blacs_intelmpi_ilp64 -lpthread -lm -ldl"
```

A useful tool for compiling applications using MKL : Intel MKL link advisor.

Intel oneAPI

Available Intel oneApi:

Version	Module
Intel oneAPI Compilers 2022	2022.0.2

Compilers:

- icc - Intel(R) C++ Compiler Classic
- icpc - Intel(R) C Compiler Classic
- ifort - Intel(R) Fortran Compiler Classic
- icx - Intel(R) oneAPI C Compiler
- icpx - Intel(R) oneAPI C++ Compiler
- ifx - Intel(R) Fortran Compiler
- dpcpp - Intel(R) oneAPI DPC++ (Data Parallel C++)

Example:

```
module load oneAPI/intel-oneapi-compilers-2022.0.2
icx --version
```

GCC

Different versions of GNU C, C++ and Fortran compilers are from Software Collections repository are available:

Version	Module
GCC 7.3.1	gcc/7
GCC 8.2.1	gcc/8
GCC 9.3.1	gcc/9

Example:

GCC 8

```
module load gcc/8
gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/opt/rh/devtoolset-8/root/usr/libexec/gcc/x86_64-redhat-linux/8/lto-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/opt/rh
/devtoolset-8/root/usr --mandir=/opt/rh/devtoolset-8/root/usr/share/man --infodir=/opt/rh/devtoolset-8/root/usr
/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-
checking=release --enable-multilib --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --
enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --
with-default-libstdcxx-abi=gcc4-compatible --enable-plugin --enable-initfini-array --with-isl=/builddir/build
/BUILD/gcc-8.2.1-20180905/obj-x86_64-redhat-linux/isl-install --disable-libmpx --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 8.2.1 20180905 (Red Hat 8.2.1-3) (GCC)
```

PGI

One version of PGI Community Edition Version 19.10 compiler is available:

Version	Module
PGI Community Edition Version 19.10	pgi/19.10

Compilers:

- pgcc - C compiler
- pgc++ - C++ compiler
- pgfortran - Fortran compiler.

Example:

PGI 19.10

```
module load pgi/19.10
pgcc --version
pgc++ --version
pgfortran --version
```

NVIDIA HPC SDK

There are multiple versions of libraries and compilers for NVIDIA GPUs. More info on :https://developer.nvidia.com/hpc-sdk. Available are:

	Version	Module
NVIDIA HPC SDK	20.7	nvhpc/20.7
	21.5	nvhpc/21.5
	22.2	nvhpc/22.2
NVIDIA HPC SDK without compilers	20.7	nvhpc-byo-compiler/20.7
	21.5	nvhpc-byo-compiler/21.5
	22.2	nvhpc-byo-compiler/22.2
NVIDIA HPC SDK without MPI	20.7	nvhpc-nompi/20.7
	21.5	nvhpc-nompi/21.5
	22.2	nvhpc-nompi/22.2

Compilers:

- nvc - C compiler
- nvc++ - C++ compiler
- nvfortran - Fortran compiler.

MPI

There are available multiple versions of MPI libraries. It is best to use those compiled with Intel compilers:

Version	Compiler	Module
MVAPICH2 2.0	GNU 4.8.5	mpi/mvapich2-2.0-x86_64 mpi/mvapich2-x86_64
MVAPICH2 2.2	GNU 4.8.5	mpi/mvapich2-2.2-x86_64
MVAPICH2 2.3	GNU 4.8.5	mpi/mvapich23-x86_64
MVAPICH2 2.0	Intel 2017	mpi/mvapich2-intel2017-2.0-x86_64 mpi/mvapich2-intel2017-x86_64
MVAPICH2 2.2	Intel 2017	mpi/mvapich2-intel2017-2.2-x86_64
MVAPICH2 2.2	Intel 2017, CUDA 9.0	mpi/mvapich2-intel2017-cuda90-2.2-x86_64
MVAPICH2 2.0	Intel 2018	mpi/mvapich2-intel2018-x86_64 mpi/mvapich2-intel2018-2.0-x86_64
MVAPICH2 2.2	Intel 2018	mpi/mvapich2-intel2018-2.2-x86_64

MVAPICH2 2.2	Intel 2018, CUDA 10.0	mpi/mvapich2-intel2018-cuda-2.2-x86_64
MVAPICH2 2.0	Intel 2019	mpi/mvapich2-intel-2.0-x86_64 mpi/mvapich2-intel-x86_64
MVAPICH2 2.2	Intel 2019	mpi/mvapich2-intel-2.2-x86_64
MVAPICH2 2.2	Intel 2019, CUDA 10.1	mpi/mvapich2-intel2019-cuda101-2.2-x86_64
OpenMPI 1.10	GNU 4.8.5	mpi/openmpi-x86_64
OpenMPI 3.0	GNU 4.8.5	mpi/openmpi3-x86_64
OpenMPI 3.1	GNU 4.8.5	mpi/openmpi31-x86_64
OpenMPI 4.1	GNU 4.8.5	mpi/openmpi41-x86_64
OpenMPI 1.10	Intel 2019	mpi/openmpi-intel-x86_64
OpenMPI 1.10	Intel 2019, CUDA 10.0	mpi/openmpi-intel-cuda-x86_64
OpenMPI 2.1	Intel 2017	openmpi21-intel-x86_64
OpenMPI 3.0	Intel 2019	mpi/openmpi3-intel-x86_64
OpenMPI 3.0	Intel 2019, CUDA 10.0	mpi/openmpi3-intel-cuda-x86_64
OpenMPI 3.1	Intel 2019	mpi/openmpi31-intel-x86_64
OpenMPI 4.1	Intel 2019	mpi/openmpi41-intel-x86_64
OpenMPI 3.1	PGI Community Edition Version 19.10	mpi/openmpi31-pgi-x86_64
Intel MPI 2021.5	Intel MPI 2021	oneAPI/intel-oneapi-mpi-2021.5.1

MVAPICH2

Example how to run an application compiled with MVAPICH2 2.2:

```
module load mpi/mvapich2-intel-2.2-x86_64
mpirun_rsh -np $NSLOTS -hostfile $TMPDIR/machines -export-all application
```

OpenMPI

Example how to run an application compiled with OpenMPI 3:

```
module load mpi/openmpi3-intel-x86_64
mpirun -np $NSLOTS -machinefile $TMPDIR/machines application
```

Intel MPI

Example how to run an application compiled with Intel MPI:

```
module load oneAPI/intel-oneapi-mpi-2021.5.1 mpirun -np $NSLOTS -hosts application
```

User application

All information on all applications maintain by Srce are available on this pages:

- bcftools
- bwa-mem2
- Gatk4
- OpenFoam
- Plink