University of Manitoba

Digital Research Alliance of Canada

Introduction to High Performance Computing Software and Lmod

Ali Kerrache

HPC Specialist

UManitoba Autumn 2026- HPC & Cloud computing workshop - Oct 21-23, 2026

University of Manitoba

Outline

- Software distribution on HPC clusters.
- How to find software packages & modules?
- Software stacks on Grex / Alliance clusters.
- Hands on:
  - How to use modules to search for installed software?
- Software installation [some examples] + modules

University of Manitoba

Software distribution on HPC clusters

Operating system package managers / repos:

- **Ubuntu:** ~$ *sudo apt-get install <package>*
- **CentOS:** ~$ *sudo yum install <package>*
- **On HPC:** users do not have **sudo**! [***NO NEED TO ASK FOR IT***]

Centralized software stack:

- **Software distributed via CVMFS:** software stacks and modules, …
- **Local software:** modules, restricted software (VASP, Gaussian, ORCA, …etc)

Local installation: usually to $HOME or $PROJECT, …

- **Download the code** [sources/binaries]: wget, git clone, curl, … etc.
- **Settings:** load dependencies, set environment variables, … etc.
- **Build:** ./configure {cmake ..} +opts; make; make test {check}; make install

University of Manitoba

Software on HPC clusters

- Home made: programs, scripts and tools, … etc.

Up to a user, … Help is available!

➔ Free Software: GNU Public License.

- ◆ Open Source, Binaries, Libraries, Compilers, Tools, …

Maintained by analysts and installed as modules.

➔ Commercial Software: restricted [VASP, STATA, … etc]

- ◆ Contact support and provide more details about the license, …
- ◆ We install the program & protect it with a POSIX group.

Maintained by analysts and admins.

University of Manitoba

Modules on HPC clusters

What is a module?

- Configuration files with instructions to modify or initialize environment variables such as PATH, LD_LIBRARY_PATH. … etc in order to use different installed programs.
- The modular architecture allows multiple versions of the same program to be installed without conflicts.

Why modules?

- Control different versions of the same program.
- Avoid conflicts between different versions and libraries.
- Set the right path to each program and library.

University of Manitoba

Lmod commands



- module **list**; module **avail**
- module **spider** <soft>/<version>
- module **load** <deps> <soft>/<version>
- module **unload {rm}** <soft>/<version>
- module **show** <soft>/<version>
- module **help** <soft>/<version>
- module **whatis** <soft>/<version>
- module **purge**; module --force **purge**
- module **use** ~/modulefiles
- module **unuse** ~/modulefiles

```
[~@bison ]$  module list
Currently Loaded Modules:
SBEnv (S)
Where:
S:  Module is Sticky, requires --force to unload or purge
```

```
[~@rorqual3: ~]$ module list

Currently Loaded Modules:
  1) CCconfig          4) gcc/12.3   (t)   7) libfabric/1.18.0
 10) openmpi/4.1.5     (m)  13) aocl-lapack/5.1
  2) gentoo/2023  (S)  5) hwloc/2.9.1     8) pmix/4.2.4
 11) flexiblas/3.3.1     14) StdEnv/2023     (S)
  3) gcccore/.12.3 (H)  6) ucx/1.14.1      9) ucc/1.2.0
 12) aocl-blas/5.1
```

University of Manitoba

Software stacks

- A set of compilers and libraries:
  - GCC, AOCC, Intel compilers, …
  - Libraries: hdf5, boost, netcdf, petsc, gsl, gdal, geos, proj, … etc.
- Modules hierarchy:
  - arch: branch for a given architecture {~~sse3~~, avx2, avx512, …}
  - CUDA: any program using GPU acceleration under the same tree.
  - Core modules: java, perl, … etc.
  - Compiler:
    - GCC: programs compiled with gcc
    - AOCC: programs compiled with gcc
    - Intel: compiled with Intel
  - OpenMPI: Compiler [Intel, GCC, AOCC] / OpenMPI
- Possibility to maintain one or more software stacks.

University of Manitoba

Software stacks on Grex

- Grex environment [default]: SBEnv
  - no module loaded by default, two architectures: avx2/avx512
  - use module spider <software name> to search for modules
  - Compilers {GCC, AOCC, Intel}, MKL, PETSc, GSL, NetCDF… etc.
  - Gaussian, ANSYS, MATLAB, … etc.
- The Alliance (CC) environment [*optional*]: CCEnv
  - Switch to CCEnv; load a standard environment; choose the architecture[avx2/avx512], use module spider <software name>

```
module load CCEnv
module load arch/avx512
module load StdEnv/2023
module load gcc/12.3 geant4/11.3.0
```

University of Manitoba

Software maintenance

Compilers/Libraries and more:

- Compilers: GCC [8.5 - 14.3]; Intel [2019, 2023], … etc.
- Libraries: HDF5, PETSc, GSL, MKL, Libxc, Boost, NetCDF, ...
- Gaussian, ANSYS, MATLAB, VASP, ORCA, MCR, Java, Python, R, … etc.
- LAMMPS, GROMACS, OpenMM, QE, OpenBABEL, … etc.

Software maintenance on Grex and Alliance clusters:

- We install programs and update modules on request from users.
- Search for a program using "module spider <name of your program>"
- If not installed, ask support "support@tech.alliancecan.ca"
- We will install the module and/or update the version.
- For commercial software, contact us before you purchase the code:
  - to check license type.
  - see if it will run under Linux environment, … etc.

University of Manitoba

Lmod summary

Useful commands for working with modules

- module **list**
- module **avail**
- module **spider** <soft>/<version>
- module **load** <soft>/<version>
- module **unload {rm}** <soft>/<version>
- module **show** <soft>
- module **help** <soft>
- module **whatis** <soft>
- module **purge;** module --force **purge**
- module **use** ~/modulefiles; module **unuse** ~/modulefiles

University of Manitoba

Modules: exercises

List of modules: python, java, perl, hdf5, netcdf, lammps, gromacs, cp2k, …

- Pick one module from the above list <myprogram>
- Run the command: module spider <myprogram>
- What to expect:
  - The module does not exist {ask for support if needed}
  - One or more versions of the module are available.
    - If many, pick one and run: module spider <myprogram>/<version>
    - Read the instructions, load the dependencies and the module
    - Experiment with other commands: module list, module show myprogram, module help myprogram, module whatis, module rm, …
- Run "module purge" and repeat the exercise for another program.
- Switch to CCEnv, then load lammps-omp and gromacs.

University of Manitoba

Quick test on login node: LAMMPS

Directory: quick-lammps-run

➔ Have a look to the scripts using cat {more or less} command:

- Grex: grex/grex-runlmp*.sh
- CC/MC: mc-mc/mc-runlmp-*sh

  Example: cat grex-runlmp-1cpu-serial.sh

➔ Run the scripts from the login node:

★ Grex:

- Serial test: sh ./grex-runlmp-1cpu-serial.sh
- OpenMP test: sh ./grex-runlmp-4cpu-openmp.sh
- MPI test: sh ./grex-runlmp-4cpu-mpi.sh

★ CC/MC:

- Serial test: sh ./mc-runlmp-1cpu-serial.sh
- OpenMP test: sh ./mc-runlmp-4cpu-openmp.sh
- MPI test: sh ./mc-runlmp-4cpu-mpi.sh

University of Manitoba

CVMFS on Grex

```
[~@bison ~]$  cvmfs_config probe
Probing /cvmfs/cvmfs-config.computecanada.ca... OK
Probing /cvmfs/soft.computecanada.ca... OK
Probing /cvmfs/restricted.computecanada.ca... OK
```

```
[~@bison ~]$  ls -1 /cvmfs/
cvmfs-config.computecanada.ca
restricted.computecanada.ca
soft.computecanada.ca
```

```
[~@~]$ module load CCEnv
[~@~]$ module load arch/avx512
[~@~]$ module load StdEnv/2023
[~@~]$ module spider geant4
[~@~]$ module spider geant4/11.3.0

[~@~]$ module load StdEnv/2023  gcc/12.3
geant4/11.3.0
```

```
[~@bison ~]$  ls /cvmfs/
cvmfs-config.computecanada.ca  restricted.computecanada.ca  soft.computecanada.ca
```

University of Manitoba

CVMFS on Grex: neurodesk.ardc.edu.au

```
[~@yak ~]$  ls /cvmfs/
cvmfs-config.computecanada.ca  restricted.computecanada.ca  soft.computecanada.ca
[~@yak ~]$ ls /cvmfs/neurodesk.ardc.edu.au
[~@yak ~]$ ls /cvmfs
      cvmfs-config.computecanada.ca  neurodesk.ardc.edu.au
      restricted.computecanada.ca  soft.computecanada.ca
[~@yak ~]$  ls /cvmfs/neurodesk.ardc.edu.au/neurodesk-modules
[~@yak ~]$  module use /cvmfs/neurodesk.ardc.edu.au/neurodesk-modules
[~@yak ~]$  module spider fsl
[~@yak ~]$  module spider functional_imaging/fsl/6.0.7.18
[~@yak ~]$  module load functional_imaging/fsl/6.0.7.18
[~@yak ~]$  module load singularity
[~@yak ~]$  module list
[~@yak ~]$  which fsl
/cvmfs/neurodesk.ardc.edu.au/containers/fsl_6.0.7.18_20250928/fsl
```

https://neurodesk.org/getting-started/neurocontainers/cvmfs/

University of Manitoba

More readings

Grex:

- ➔ https://um-grex.github.io/docs/software/using-modules/
- ➔ https://um-grex.github.io/docs/software/software-list/

Alliance clusters:

- ➔ https://docs.alliancecan.ca/wiki/Utiliser_des_modules/en
- ➔ https://docs.alliancecan.ca/wiki/Available_software

Modules:

- ➔ https://lmod.readthedocs.io/en/latest/010_user.html
- ➔ https://modules.readthedocs.io/en/v4.1.3/index.html
- ➔ https://lmod.readthedocs.io/en/stable/015_writing_modules.html



University of Manitoba



Digital Research Alliance of Canada

Building software

University of Manitoba

Building software

- Local installation [user's directory: home, project]:
  - R packages; Julia packages, Perl modules
  - Python packages: virtual environment
  - Home made programs and commercial software.

- Installation with:
  - make; make test {check}; make install
  - configure; make; make test {check}; make install
  - cmake; make; make test {check}; make install

- Java applications: jar files
- Containers: Singularity, Aptainer, Podman, Pixis {*separate talks*}
  - build the image and run your program using the container

University of Manitoba

What programs are installed locally?

- R packages: minimal installation
  - R as a module: users can install the packages in their home directory.
- Python as a module: python and scipy-stack or scipy-bundle
  - users can install the packages needed in their home or project directories.
- Julia as a module: packages installed locally
- Perl as a module:
  - users can install the packages needed in their home directory.
- Other software installed locally:
  - Home made programs {up to a user or a group}
  - Restricted and licensed software that can not be distributed
  - Custom software: patch from a user, changing parts of the code, … etc.

University of Manitoba

Build software using: configure/make

- Download and unpack the code: wget, … gunzip, … etc.
- Load a compiler and dependencies: module load gcc openmpi fftw
- Configure the program
  - If configure not included, run: autoreconf -fvi [to generate it].
  - ./configure --help [to see the different options].
  - ./configure --prefix=<path to install dir> {+other options}
- Compile and test:
  - make; make -j4
  - make check; make test
- Install the program:
  - make install
- Set the path: export PATH=${PATH}:<path to install dir>/bon

University of Manitoba

Build software using: cmake/make

- Download and unpack the code: wget, … gunzip, … etc.
- Load a compiler & dependencies: module load gcc ompi fftw cmake
- Configure the program: you may need to load cmake module
  - mkdir build && cd build
  - cmake .. --help [to see the different options].
  - cmake .. -DCMAKE_INSTALL_PREFIX=installdir {+options}
- Compile and test:
  - make; make -j8
  - make check; make test
- Install the program:
  - make install

University of Manitoba

Hands on: software installation

- Install R packages: sp and dplyr
- Install python packages: cutadapt
- Install Perl packages: Hash::Merge
- Install Julia packages: JLD (requires HDF5)
- Install Java applications: Trimmomatic
- Install STAR: uses make (Makefile included in the archive)
- Install Treemix: uses configure to generate the Makefile
  ./configure {+opts} && make && make test {check} && make install
- Install DIAMOND: uses cmake to generate the Makefile
  mkdir build && cd build && cmake .. {+opts} && make -j4 && make install
- Install Geant4: uses cmake to generate the Makefile
  mkdir build && cd build && cmake .. {+opts} && make -j4 && make install

University of Manitoba

Local installation: R packages

- ➔ R packages: rgdal, adegenet, stats, rjags, dplyr, sf, … etc.
- ➔ Choose a module version: module spider r
- ➔ Load R and dependencies (gdal, geos, jags, gsl, udunits… etc):
  - ◆ module load gcc r <+other external modules>
- ➔ Launch R and install the packages:

```
~$ R
> install.packages(”dplyr")
'lib =/cvmfs/soft.computecanada.ca/easybuild/{..}/R/library"' is not writable
Would you like to use a personal library instead? (yes/No/cancel) yes
Would you like to create a personal library ‘~/R/{…}’ to install packages into? (yes/No/cancel) yes
--- Please select a CRAN mirror for use in this session —
> install.packages(”other packages")
```

University of Manitoba

Local installation: Python packages

- Load the modules:
  - module load <deps> python <+ext>
- Create a virtual environment
  - virtualenv ~/my_venv
- Activate the virtual environment
  - source ~/my_venv/bin/activate
- Update pip
  - pip install --no-index --upgrade pip
- Install the packages
  - pip install pandas
  - pip install -r requirements.txt
  - ~~python setup.py install~~

```
module load gcc python
virtualenv ~/my_venv
source ~/my_venv/bin/activate
pip install cutadapt
pip install <other package>
pip freeze > ~/requirements.txt
deactivate
```

```
source ~/my_venv/bin/activate
cutadapt [+options]
deactivate
```

University of Manitoba

Local installation: Perl packages

- Example: Hash::Merge; Logger::Simple; MCE::Mutex; threads …
- Load Perl module: module load perl
- Install the the first package using cpan or cpanm:
  - ~$ cpan install YAML

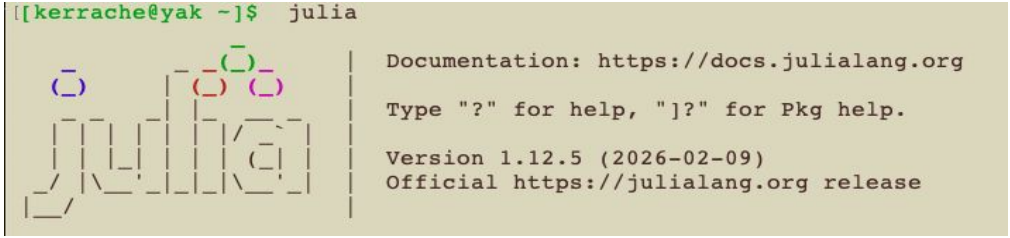
  Would you like to configure as much as possible automatically? [yes] **yes**
  What approach do you want? (Choose 'local::lib', 'sudo' or 'manual') [local::lib] **local::lib**
  Would you like me to append that to /home/$USER/.bashrc now? [yes] **yes**
- Install the rest of the packages using cpan or cpanm:
  - ~$ cpan install Hash::Merge
  - ~$ cpan install Logger::Simple
  - ~$ cpan install MCE::Mutex
- Update the ~/.bashrc [for the first time, it asks to update ~/.bashrc]

University of Manitoba

Local installation: Julia packages

JLD: Save and load variables in Julia Data format

```
[~@yak ~]$ module load arch/avx512  gcc/13.2.0 hdf5/1.14.6 julia
[~@yak ~]$ export HDF5_DIR=$MODULE_HDF5_PREFIX
```




```
[~@yak ~]$  julia

julia> using Libdl
julia> push!(Libdl.DL_LOAD_PATH, ENV["HDF5_DIR"] * "/lib")
julia> using Pkg
julia> Pkg.add("JLD")
julia> using JLD
```

https://docs.alliancecan.ca/wiki/Julia#Installing_packages

University of Manitoba

Java applications

- Download and unpack the code using wget, curl, … etc.
- Load java module [run module spider java]
- Run the code.

- Example: Trimmomatic

  wget https://github.com/usadellab/Trimmomatic/releases/download/v0.40/Trimmomatic-0.40.zip

  unzip Trimmomatic-0.40.zip

- Run the code
  - Load java module: module load openjdk
  - java -jar <path to>/trimmomatic-0.40.jar {+options if any}

University of Manitoba

STAR

- Download the code {wget; curl; git clone; …}:

  wget https://github.com/alexdobin/STAR/archive/refs/tags/2.7.11b.tar.gz

- Unpack the code: tar -xvf 2.7.11b.tar.gz
- Load GCC compiler: module load gcc
- Compile the code:

  cd STAR-2.7.11b/source

  make

- Copy the binaries and set the path:

  mkdir -p ~/software/star/2.7.11b/bin

  cp STAR ~/software/star/2.7.11b/bin

  export PATH=$PATH:${HOME}/software/star/2.7.11b/bin

University of Manitoba

Treemix

- Download the source files:
  wget https://bitbucket.org/nygcresearch/treemix/downloads/treemix-1.13.tar.gz
- Unpack the source files: tar -xvf treemix-1.13.tar.gz
- Change the directory: cd treemix-1.13/
- Load the modules: module load gcc boost
- Configure: ./configure --prefix=/home/$USER/software/treemix/1.13
- Compile and install: make && make test && make install
- Set a path: export PATH=$PATH:$HOME/software/treemix/1.13/bin
- Usage in a job script:

```
module load gcc boost
export PATH=$PATH:$HOME/software/treemix/1.13/bin
treemix {+options if any}
```

University of Manitoba

DIAMOND

- Download the source files:
  wget https://github.com/bbuchfink/diamond/archive/refs/tags/v2.2.8.tar.gz
- Unpack the source files: tar -xvf v2.2.8.tar.gz
- Change the directory: cd diamond-2.2.8/
- Load the modules: module load gcc cmake
- mkdir build && cd build
- Configure: cmake .. --DCMAKE_INSTALL_PREFIX=${HOME}/Softs/diamond/2.1.25
- Compile and install: make && make test && make install
- Set a path: export PATH=$PATH:$HOME/Softs/diamond/2.1.25/bin
- Usage in a job script:

```
module load gcc boost
export PATH=$PATH:$HOME/Softs/diamond/2.2.8/bin
diamond {+options if any}
```

University of Manitoba

Geant4

- Download the source files:
  wget
  https://gitlab.cern.ch/geant4/geant4/-/archive/v11.3.1/geant4-v11.3.1.tar.gz
- Unpack the source files: tar -xvf geant4-v11.3.1.tar.gz
- Change the directory: cd geant4-v11.3.1/
- Load the modules: gcc clhep tbb qt cmake …
- mkdir build && cd build
- Create a configure.cmd file {Link}
- source ./configure.cmd
- make
- make test
- make install

University of Manitoba

Content of configure.cmd for Grex

```
cmake .. -DCMAKE_INSTALL_PREFIX=${HOME}/Softs/geant4/11.3.1 -DCMAKE_BUILD_TYPE="Release"
-DGEANT4_BUILD_MULTITHREADED="ON" -DBUILD_STATIC_LIBS:BOOL="ON"
-DGEANT4_USE_GDML:BOOL="ON" -DXercesC_LIBRARY_RELEASE="/usr/lib64/libxerces-c.so"
-DEXPAT_LIBRARY="/usr/lib64/libexpat.so" -DEXPAT_INCLUDE_DIR="/usr/include"
-DCLHEP_DIR="$MODULE_CLHEP_PREFIX/lib/CLHEP-${CLHEPVERSION}"
-DPKG_CONFIG_EXECUTABLE="/usr/bin/pkg-config" -DGEANT4_BUILD_STORE_TRAJECTORY:BOOL="ON"
-DGEANT4_BUILD_TESTS:BOOL="ON" -DGEANT4_USE_RAYTRACER_X11="ON"
-DGEANT4_USE_SYSTEM_ZLIB="ON" -DGEANT4_USE_OPENGL_X11="ON"
-DGEANT4_ENABLE_TESTING="ON" -DGEANT4_USE_TBB="ON" -DGEANT4_USE_QT:BOOL="OFF"
-DQT_QMAKE_EXECUTABLE="$MODULE_QT_PREFIX/bin/qmake" -DGEANT4_USE_QT_QT6="ON"
-DGEANT4_INSTALL_EXAMPLES:BOOL="ON" -DGEANT4_INSTALL_DATA:BOOL="ON"
-DGEANT4_INSTALL_DATASETS_NUDEXLIB="ON" -DGEANT4_INSTALL_DATASETS_TENDL="ON"
-DGEANT4_INSTALL_DATASETS_URRPT="ON" -DGEANT4_USE_HDF5="OFF" -DGEANT4_USE_VTK="OFF"
-DGEANT4_USE_FREETYPE="ON" -DGEANT4_USE_G3TOG4="ON" -DGEANT4_USE_GDML="ON"
-DGEANT4_USE_SMARTSTACK="ON" -DGEANT4_USE_INVENTOR="OFF" -DGEANT4_USE_PTL_LOCKS="ON"
-DGEANT4_USE_SMARTSTACK="ON" -DGEANT4_USE_SYSTEM_PTL="OFF" -DGEANT4_USE_USOLIDS="OFF"
-DGEANT4_USE_XM="ON"
```

University of Manitoba

Content of configure.cmd for CC

```
cmake .. -DCMAKE_INSTALL_PREFIX=${HOME}/Softs/geant4/11.3.1 -DCMAKE_BUILD_TYPE="Release"
-DGEANT4_BUILD_MULTITHREADED="ON" -DBUILD_STATIC_LIBS:BOOL="ON"
-DGEANT4_USE_GDML:BOOL="ON" -DXercesC_LIBRARY_RELEASE="$EBROOTGENTOO/lib64/libxerces-c.so"
-DEXPAT_LIBRARY="$EBROOTGENTOO/lib64/libexpat.so"
-DEXPAT_INCLUDE_DIR="$EBROOTGENTOO/include"
-DCLHEP_DIR="$EBROOTCLHEP/lib/CLHEP-${CLHEPVERSION}"
-DPKG_CONFIG_EXECUTABLE="/usr/bin/pkg-config" -DGEANT4_BUILD_STORE_TRAJECTORY:BOOL="ON"
-DGEANT4_BUILD_TESTS:BOOL="ON" -DGEANT4_USE_RAYTRACER_X11="ON"
-DGEANT4_USE_SYSTEM_ZLIB="ON" -DGEANT4_USE_OPENGL_X11="ON"
-DGEANT4_ENABLE_TESTING="ON" -DGEANT4_USE_TBB="ON" -DGEANT4_USE_QT:BOOL="OFF"
-DQT_QMAKE_EXECUTABLE="$EBROOTQT/bin/qmake" -DGEANT4_USE_QT_QT6="ON"
-DGEANT4_INSTALL_EXAMPLES:BOOL="ON" -DGEANT4_INSTALL_DATA:BOOL="ON"
-DGEANT4_INSTALL_DATASETS_NUDEXLIB="ON" -DGEANT4_INSTALL_DATASETS_TENDL="ON"
-DGEANT4_INSTALL_DATASETS_URRPT="ON" -DGEANT4_USE_HDF5="OFF" -DGEANT4_USE_VTK="OFF"
-DGEANT4_USE_FREETYPE="ON" -DGEANT4_USE_G3TOG4="ON" -DGEANT4_USE_GDML="ON"
-DGEANT4_USE_SMARTSTACK="ON" -DGEANT4_USE_INVENTOR="OFF"
-DGEANT4_USE_PTL_LOCKS="ON" -DGEANT4_USE_SMARTSTACK="ON"
-DGEANT4_USE_SYSTEM_PTL="OFF" -DGEANT4_USE_USOLIDS="OFF"  -DGEANT4_USE_XM="ON"
```



University of Manitoba



Digital Research Alliance of Canada

Thank you for your attention

Any question?

University of Manitoba

ABINIT

```
module load intel openmpi gsl netcdf
instdir=<path to the installation directory>

../configure --prefix=${instdir} --enable-mpi --enable-mpi-io --with-fft-flavor=fftw3-mkl
--with-linalg-flavor=mkl --with-math-flavor=gsl --enable-debug="no"
--enable-optim="standard" --enable-64bit-flags
--with-linalg-libs="-L$MKLROOT/lib/intel64 -lmkl_scalapack_lp64
-lmkl_blacs_openmpi_lp64  -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lm"
--with-fft-incs="-I$MKLROOT/include/fftw -I$MKLROOT/interfaces/fftw3xf"
--with-fft-libs="-L$MKLROOT/interfaces/fftw3xf -lfftw3xf_intel_lp64"
--with-dft-flavor="atompaw+libxc+wannier90" --with-trio-flavor="netcdf"  --enable-lotf
--enable-macroave   --enable-gw-dpc  CC=mpicc CXX=mpic++ FC=mpif90
F77=mpif77 F90=mpif90
```



University of Manitoba

PETSc

```
./configure --with-blas-lapack-dir=$MKLROOT/lib/intel64  --prefix=${instdir}
--with-cxx-dialect=C++11 --download-scalapack=yes --download-blacs=yes
--download-superlu_dist=yes --download-mumps=yes --download-parmetis=yes
--download-metis=yes --download-spooles=yes   --download-cproto=yes
--download-prometheus=yes --with-mkl_pardiso=1 --with-mkl_pardiso-dir=$MKLROOT
--with-mkl-sparse-optimize=1   --with-scalar-type=complex  --with-debugging=0  --with-hdf5=yes
--with-hdf5-dir=$HDF5HOME  --download-suitesparse=yes   --download-fftw=${fftsrc}
--download-amd=yes --download-adifor=yes  --download-superlu=yes --download-triangle=yes
--download-generator=yes   --with-64-bit-pointers=no  --with-cc=mpicc  --CFLAGS='-O2
-I$MKLROOT/include -mkl -fPIC '   --with-cxx='mpicxx'  --CXXFLAGS='-O2 -I$MKLROOT/include
-mkl -std=c++11 -fPIC '  --with-fc='mpif90'  --FFLAGS='-O2 -I$MKLROOT/include -mkl -fPIC '
--with-single-library=yes --with-shared-libraries=yes  --with-shared-ld=mpicc
--sharedLibraryFlags="-fpic -mkl -fPIC"  --with-mpi=yes  --with-mpi-shared=yes
--with-mpirun=mpiexec  --with-mpi-compilers=yes  --with-x=yes {+other options}
make && make install
```

University of Manitoba

GROMACS

- Download and unpack the source files
- Load modules:

  module load intel openmpi fftw cmake
- configure; compile; install

  cd gromacs-5.1.4; mkdir build; cd build

  cmake -DCMAKE_INSTALL_PREFIX=<path to install dir> -DBUILD_SHARED_LIBS=off -DBUILD_TESTING=off -DREGRESSIONTEST_DOWNLOAD=off -DCMAKE_C_COMPILER=`which mpicc` -DCMAKE_CXX_COMPILER=`which mpicxx` -DGMX_BUILD_OWN_FFTW=on -DGMX_SIMD=SSE4.1 -DGMX_DOUBLE=off -DGMX_EXTERNAL_BLAS=on -DGMX_EXTERNAL_LAPACK=on -DGMX_FFT_LIBRARY=fftw3 -DGMX_GPU=off -DGMX_MPI=on -DGMX_OPENMP=off -DGMX_X11=on ../gromacs-5.1.4

  make -j4; make install

University of Manitoba

Summary

- Use Lmod commands to search for the modules:
  - Compilers, OpenMPI, NetCDF, HDF5, PETSc, Gaussian, ANSYS, MATLAB, ORCA, MCR, Java, Python, R, … etc.
- Some packages require a local installation:
  - Home made programs, Python, Perl, R, Julia packages, ...
- Software maintenance on Grex and Alliance clusters:
  - Search for a program using “module spider <name of your program>”
  - If not installed, ask for support “support@tech.alliancecan.ca”
  - We will install the module and/or update the version.
  - For commercial software, contact us before you purchase the code:
    - to check license type.
    - see if it will run under Linux environment, … etc.



University of Manitoba



Digital Research Alliance of Canada

Additional Slides

University of Manitoba

module list: narval

[~@narval2: ~]$ module list
Currently Loaded Modules:

1) CCconfig	5) hwloc/2.9.1	9) ucc/1.2.0	13) StdEnv/2023 (S)
2) gentoo/2023 (S)	6) ucx/1.14.1	10) openmpi/4.1.5 (m)	14) mii/1.1.2
3) gcccore/.12.3 (H)	7) libfabric/1.18.0	11) flexiblas/3.3.1	
4) gcc/12.3 (t)	8) pmix/4.2.4	12) blis/0.9.0	

Where:

- S: Module is Sticky, requires --force to unload or purge
- m: MPI implementations / Implémentations MPI
- t: Tools for development / Outils de développement
- H: Hidden Module

University of Manitoba

module list: Grex

[~@bison ~]$ module list

Currently Loaded Modules:
1) SBEnv (S)

CC software stack:
[~@bison ~]$ module load CCEnv
[~@bison ~]$ module load arch/avx512
[~@bison ~]$ module load StdEnv/2023

Where:
S: Module is Sticky, requires --force to unload or purge

Note:

- Before starting, make sure you have the appropriate software stack and the compilers and libraries you need.
- Use “module spider” to search for the programs.

University of Manitoba

Module example: python

```
[~@bison: ~]$ module spider python
python:
   Description:
      The Python programming language. Homepage: https://www.python.org/
   Versions:
      python/3.10.14
      python/3.10.16
      python/3.11.8
      python/3.11.11
      python/3.12.9
  For detailed information about a specific "python" package (including how to load the modules) use the module's full
name.
  Note that names that have a trailing (E) are extensions provided by other modules.
  For example:
      $ module spider python/3.12.9
```

University of Manitoba

Module example: python/3.12.9

[~@bison: ~]$ module spider python/3.12.9

python: python/3.12.9

Description:

The Python programming language. Homepage: https://www.python.org/

You will need to load all module(s) on any one of the lines below before the "python/3.12.9" module is available to load.

arch/avx512 gcc/13.2.0
arch/avx512 gcc/14.3.0
arch/avx512 intel-one/2024.1
cuda/12.4.1 arch/avx2 gcc/13.2.0

Help:

The Python programming language.
Homepage: https://www.python.org/

Use on of the following to load the module:

module load arch/avx512 gcc/13.2.0 python/3.12.9
module load arch/avx512 gcc/14.3.0 python/3.12.9
module load arch/avx512 intel-one/2024.1 python/3.12.9
module load cuda/12.4.1 arch/avx2 gcc/13.2.0 python/3.12.9

University of Manitoba

module show python/3.12.9

```
[~@bison: ~]$ module show python/3.12.9
/global/software/alma8/sb/modules/base/python/3.11.8:
prepend_path("CMAKE_PREFIX_PATH","/global/software/alma8/sb/opt/base/python/3.11.8")
prepend_path("PATH","/global/software/alma8/sb/opt/base/python/3.11.8/bin")
prepend_path("LD_LIBRARY_PATH","/global/software/alma8/sb/opt/base/python/3.11.8/lib")
prepend_path("MANPATH","/global/software/alma8/sb/opt/base/python/3.11.8/share/man")
setenv("MODULE_PYTHON_PREFIX","/global/software/alma8/sb/opt/base/python/3.11.8")
prepend_path("MODULEPATH","/global/software/alma8/sb/modules/python-3.11.8")
whatis("Description: The Python programming language. Homepage: https://www.python.org/")
family("python")
setenv("PIP_DISABLE_PIP_VERSION_CHECK","1")
setenv("PYDEVD_DISABLE_FILE_VALIDATION","1")
prepend_path("PYTHONPATH","/global/software/alma8/sb/python3/site-packages")
help([[
The Python programming language.
Homepage: https://www.python.org/
]])
```

University of Manitoba

Module example: boost

```
[~@bison: ~]$ module spider boost
boost:
 Description:
      Boost provides free peer-reviewed portable C++ source libraries, emphasizing libraries that work well
      with the C++ Standard Library
 Versions:
      boost/1.78.0
      boost/1.85.0
      Other possible modules matches:
      xgboost
To find other possible module matches execute:
     $ module -r spider '.*boost.*'
For detailed information about a specific "boost" package (including how to load the modules) use the module's full name.
Note that names that have a trailing (E) are extensions provided by other modules.
For example:
$ module spider boost/1.85.0
```

University of Manitoba

Module example: boost

[~@bison: ~]$ module spider boost/1.85.0

boost: boost/1.85.0

Description:

Boost provides free peer-reviewed portable C++ source libraries, emphasizing libraries that work well with the C++ Standard Library

You will need to load all module(s) on any one of the lines below before the "boost/1.85.0" module is available to load.

arch/avx512 gcc/13.2.0

cuda/12.4.1 arch/avx2 gcc/13.2.0

- module load arch/avx512 gcc/13.2.0 boost/1.85.0
- module load cuda/12.4.1 arch/avx2 gcc/13.2.0 boost/1.85.0

Help:

Boost provides free peer-reviewed portable C++ source libraries, emphasizing libraries that work well with the C++ Standard Library. Boost libraries are intended to be widely useful, and usable across a broad spectrum of applications. The Boost license encourages both commercial and non-commercial use.

Homepage: http://www.boost.org

University of Manitoba

Module example: boost

```
[~@bison: ~]$ module load arch/avx512  gcc/13.2.0 boost/1.85.0
[~@bison: ~]$ module show boost
/global/software/alma8/sb/modules/arch-avx512-gcc-13.2.0/boost/1.85.0:
prepend_path("CMAKE_PREFIX_PATH","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0")
prepend_path("CPATH","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0/include")
prepend_path("LIBRARY_PATH","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0/lib")
prepend_path("LD_LIBRARY_PATH","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0/lib")
setenv("MODULE_BOOST_PREFIX","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0")
whatis("Description: Boost provides free peer-reviewed portable C++ source libraries, emphasizing libraries
that work well with the C++ Standard Library")
whatis("Homepage: http://www.boost.org")
setenv("BOOST_ROOT","/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0/boost/1.85.0")
help([[
    Boost provides free peer-reviewed portable C++ source libraries, emphasizing libraries that work well with the C++ Standard Library.
    Homepage: http://www.boost.org
]])
```

University of Manitoba

Module example: gromacs

```
[~@bison: ~]$ module spider gromacs
  gromacs:
     Description:
          GROMACS (GROningen MAchine for Chemical Simulations) is a molecular dynamics package primarily
          designed for simulations of proteins, lipids and nucleic acids
    Versions:
      gromacs/2022
      gromacs/2023.3
      gromacs/2024.1
      gromacs/2025.2
      gromacs/2025.3
 For detailed information about a specific "gromacs" package (including how to load the modules) use the module's full
name.
 Note that names that have a trailing (E) are extensions provided by other modules.
 For example:
    $ module spider gromacs/2025.3
```

University of Manitoba

Module example: gromacs

[~@bison: ~]$ module spider gromacs/2025.3

gromacs: gromacs/2025.3

Description:

GROMACS (GROningen MAchine for Chemical Simulations) is a molecular dynamics package primarily designed for simulations of proteins, lipids and nucleic acids

You will need to load all module(s) on any one of the lines below before the "gromacs/2025.3" module is available to load.

arch/avx512 gcc/13.2.0 openmpi/5.0.6

cuda/12.4.1 arch/avx2 gcc/13.2.0

Help:

GROMACS (GROningen MAchine for Chemical Simulations) is a molecular dynamics package primarily designed for simulations of proteins, lipids and nucleic acids. It was originally developed in the Biophysical Chemistry department of University of Groningen, and is now maintained by contributors in universities and research centers across the world. Starting from version 4.6, GROMACS is released under the GNU Lesser General Public License.

Homepage: http://www.gromacs.org

University of Manitoba

Find and load Gaussian

Gaussian: restricted software; requires a registration
https://um-grex.github.io/docs/specific-soft/gaussian/

```
[~@bison ~]$  module spider gaussian
  gaussian:
        Versions:
        gaussian/g16.b01
        gaussian/g16.c01
For detailed information about a specific "gaussian" package (including how to load the modules) use the module's full
name. Note that names that have a trailing (E) are extensions provided by other modules.
  For example:
        $ module spider gaussian/g16.c01
```

➔ [~@bison ~]$ module load gaussian
➔ [~@bison ~]$ module load gaussian/g16.c01
➔ [~@bison ~]$ module load gaussian/g16.b01

University of Manitoba

Find and load ORCA

ORCA: restricted software; requires a registration

https://um-grex.github.io/grex-docs/specific-soft/orca/

```
[~@bison ~]$ module spider orca
  orca:
        Versions:
         orca/5.0.4
         orca/6.0.1
  For detailed information about a specific "orca" package (including how to load the modules) use the
  module's full name. Note that names that have a trailing (E) are extensions provided by other modules.
  For example:
        $ module spider orca/6.0.1
➔       [~@bison ~]$ module load arch/avx512 intel/2023.2  openmpi/4.1.6 orca/5.0.4
➔       [~@bison ~]$ module load arch/avx512 gcc/13.2.0  openmpi/4.1.6 orca/6.0.1
[~@bison ~]$  module load arch/avx512  gcc/13.2.0  openmpi/4.1.6 orca/6.0.1
      To execute ORCA, run:
          ${MODULE_ORCA_PREFIX}/orca orca.inp > orca.out
```

University of Manitoba

Find and load LAMMPS

```
[~@bison ~]$ module spider lammps
  lammps:
       Versions:
            lammps/2021-09-29
            lammps/2024-08-29p1-nep
            lammps/2024-08-29p1
```

```
[~@bison ~]$ module spider lammps/2024-08-29p1
  You will need to load all module(s) on any one of the
  lines below before the "lammps/2024-08-29p1"
  module is available to load.
```

- arch/avx512 gcc/13.2.0 openmpi/4.1.6
- arch/avx512 intel-one/2024.1 openmpi/4.1.6
- cuda/12.4.1 arch/avx2 gcc/13.2.0 openmpi/4.1.6

```
[~@bison ~]$ module load arch/avx512 gcc/13.2.0 openmpi/4.1.6 lammps/2024-08-29p1
[~@bison ~]$ module list
Currently Loaded Modules:
  1) SBEnv      (S)   4) openmpi/4.1.6   7) fftw/3.3.10        10) zstd/1.5.6     13) gsl/2.7
  2) arch/avx512      5) kim/2.3.0       8) openblas/0.3.26    11) hdf5/1.14.2    14) lammps/2024-08-29p1
  3) gcc/13.2.0       6) ffmpeg/7.0.2    9) eigen/3.4.0        12) netcdf/4.9.2+hdf5-1.14.2
[~@bison ~]$ which lmp
/global/software/alma8/sb/opt/arch-avx512-gcc-13.2.0-openmpi-4.1.6/lammps/2024-08-29p1/bin/lmp
```

University of Manitoba

Find and load QE

```
[~@bison ~]$ module spider espresso
  espresso:
        Versions:
        espresso/7.3.1+aocl-4.2.0
        espresso/7.3.1
        espresso/7.4.1+aocl-4.2.0
        espresso/7.4.1
        espresso/7.5+aocl-4.2.0
        espresso/7.5
For detailed information about a specific "espresso" package (including how to load the modules) use the
module's full name. Note that names that have a trailing (E) are extensions provided by other modules.
  For example:
        $ module spider espresso/7.4.1
```

➔ [~@bison ~]$ module spider espresso/7.4.1

➔ [~@bison ~]$ module load arch/avx512 intel/2023.2 openmpi/4.1.6 espresso/7.4.1

University of Manitoba

Find and load MATLAB

```
[~@bison ~]$ module spider matlab
  matlab:
     Versions:
          matlab/R2020B2
          matlab/R2022A
          matlab/R2023B
          matlab/R2024A
   For detailed information about a specific "matlab" package (including how to load the modules) use the
module's full name.
   Note that names that have a trailing (E) are extensions provided by other modules.
   For example:
     $ module spider matlab/R2024A
```

➔ [~@bison ~]$ module load matlab/R2024A
➔ [~@bison ~]$ module load matlab/R2023B
➔ [~@bison ~]$ module load matlab/R2022A
➔ [~@bison ~]$ module load matlab/R2020B2

University of Manitoba

Find and load MCR

```
[~@bison ~]$ module spider mcr
  mcr:
     Versions:
        mcr/R2020b
        mcr/R2022a
        mcr/R2023b
        mcr/R2024a
   For detailed information about a specific "mcr" package (including how to load the modules)
   use the module's full name.

   For example:
     $ module spider mcr/R2024a
```

➔ [~@bison ~]$ module load mcr/R2020b
➔ [~@bison ~]$ module load mcr/R2022a
➔ [~@bison ~]$ module load mcr/R2023b
➔ [~@bison ~]$ module load mcr/R2024a

University of Manitoba

Modules on Grex

```
------------------------------------------------------------ /global/software/alma8/sb/modules/base ------------------------------------------------------------
   adf/2019.305-impi              eigen/3.4.0              julia/1.10.3             nodejs/20.16.0                ramalama/0.12.0
   adf/2021.106-impi              expect/5.45.4            julia/1.11.3      (D)    nodejs/20.18.1                ratarmount/1.2.0
   adf/2021.107-impi              fastqc/0.12.1            kitops/1.6.0             nodejs/22.4.1                 rclone/1.70.3
   adf/2023.104-impi              feko/2021.2              libaec/1.0.6             nodejs/22.5.1                 rclone/1.71.0          (D)
   adf/2024.105-impi-aocl         ffmpeg/7.0.2             libvori/220621           nodejs/22.11.0         (D)    rs-server/2025.05.1-513
   adf/2024.105-impi              fftw/3.3.10              libxml2/2.11.9           nvtop/3.1.0                   rs-server/2025.09.0-387 (D)
   adf/2025.104-impi-aocl    (D)  flex/2.6.4               mathematica/14.2         nvtop/3.2.0            (D)    rust/1.87.0
   admixture/1.3.0                freeglut/3.4.0           matlab-proxy/0.26.0      omlmd/0.1.6                   rust/1.89.0            (D)
   ansys/21.1                     gaussian/g16.b01         matlab-proxy/0.27.1 (D)  openeye/2022.2.1              samtools/1.20
   ansys/2023R2              (D)  gaussian/g16.c01    (D)  matlab/R2020B2           openjdk/11.0.22               scons/3.1.2
   ant/1.10.15                    gcc/8.5.0                matlab/R2022A            openjdk/17.0.11_9             singularity/4.1.2
   arch/avx2                      git-annex/10.20250828    matlab/R2023B            openjdk/17.0.12_7             singularity/4.2.2      (D)
   arch/avx512               (D)  git-lfs/3.7.0            matlab/R2024A      (D)   openjdk/17.0.13_11            skopeo/1.20.0
   autotools/2022a                git/2.49.0               mcr/R2020b               openjdk/21.0.2                snp-sites/2.5.1
   bamtools/2.5.2                 git/2.51.0          (D)  mcr/R2022a               openjdk/21.0.3_9              snpeff/5.2f
   beagle/5.4-20241029            globus/3.35.2            mcr/R2023b               openjdk/21.0.4_7              sparsehash/2.0.4
   birch/3.90                     globus/3.36.0       (D)  mcr/R2024a         (D)   openjdk/21.0.5_11             sqlite/3.35.5
   birch/4.0                 (D)  gmp/6.2.1                megahit/1.2.9            openjdk/21.0.6_7       (D)    stata/15.0-fagfs
   buildah/1.41.0                 gmp/6.3.0           (D)  metashape-pro/2.1.4      openssl/3.4.0                 stata/18.0-ffin        (D)
   busco/5.8.3                    gnina/1.1                metashape-pro/2.2.1      openstack-client/8.2.0        structure/2.3.4
   cfitsio/4.4.1                  gnina/2024          (D)  metashape-pro/2.2.2 (D)  ovito/3.12.0                  subread/2.0.8
   cfitsio/4.5.0             (D)  gnuplot/5.4.2           micro/2.0.14             ovito/3.12.3                  subversion/1.14.3
   cmake/3.28.4                   gnuplot/6.0.2       (D)  mii/1.1.1                ovito/3.13.1           (D)    swig/4.2.0
   cmake/3.31.1                   golang/1.24.4            minimap2/2.28            pandoc/3.6.1                  trimmomatic/0.39
   cmake/4.1.1               (D)  golang/1.25.1      (D)  molden/7.3               paraview-offscreen/5.10.1     unrar/7.10.2
   code-server/4.103.1            haploview/4.2            mpfr/4.2.1               picard/3.3.0                  vaspkit/1.5.1
   compleasm/0.2.6                hmmer/3.4                mumax3/3.10              podman-compose/1.5.0          velvet/1.2.10
   cppzmq/4.10.0                  htgettoken/2.0-2         mummer/4.0.0rc1          podman-tui/1.7.0              vep/113.4
   cuda/11.8.0                    intel-one/2023.2         nbo/nbo7-2021            podman/5.6.0                  vim/9.1
   cuda/12.2.2                    intel-one/2024.1    (D)  nextflow/24.10.0         podman/5.6.1           (D)    visit/3.4.2
   cuda/12.4.1               (D)  intel/2023.2             ninja/1.10.2             process-compose/1.75.1        voroplusplus/0.4.6
   cudnn/8.8.1.3+cuda-11.8.0      intelmpi/2019.8          nodejs/18.20.2           prodigal/2.6.3                wine/10.0
   deepvariant/1.8.0-gpu          intelmpi/2021.10    (D)  nodejs/18.20.4           python/3.11.8                 xz/5.6.4
   deepvariant/1.8.0         (D)  jellyfish/2.3.1         nodejs/18.20.5           qiime2/2024.10                yaml-cpp/0.8.0
   dejagnu/1.6.3                  jemalloc/5.3.0           nodejs/20.12.2           quast/5.3.0                   z3/4.13.4
   diamond/2.1.10                 jq/1.7                   nodejs/20.15.1           r/4.3.3                       zstd/1.5.6

------------------------------------------------------------------ /opt/lmod/stacks -------------------------------------------------------------------
   CCEnv (S)    SBEnv (S,L)

---------------------------------- This is a list of module extensions. Use "module --nx avail ..." to not show extensions. ----------------------------------
    autoconf (E)     automake (E)     bcftools (E)     gettext (E)     htslib (E)     java (E)     libtool (E)

These extensions cannot be loaded directly, use "module spider extension_name" for more information.
```

module avail

module spider python
module spider java
module spider gromacs

module load arch/avx512
gcc/13.2.0 openmpi/5.0.6
module avail

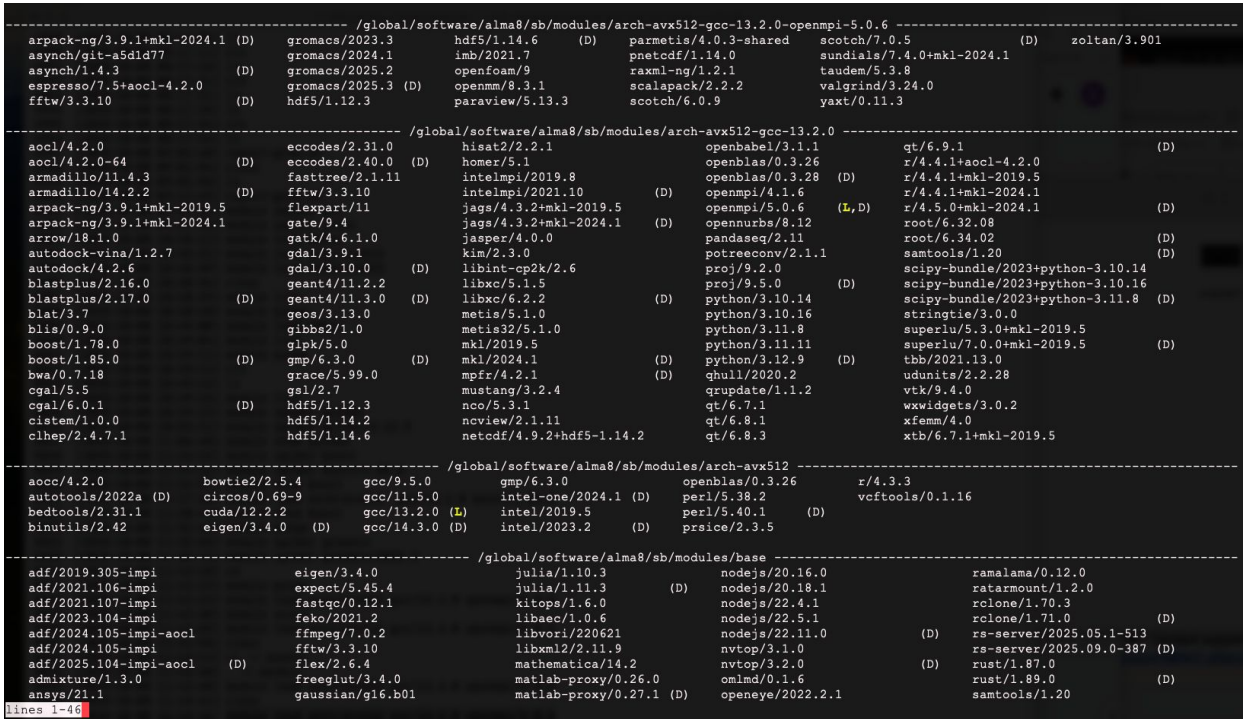
module spider <soft>
module spider <soft>/<ver>

module help <soft>
module show <soft>

module purge

University of Manitoba

Modules on Grex



module load arch/avx512
gcc/13.2.0 openmpi/5.0.6

module avail

module spider gromacs
module spider cp2k

If not available:
➔ contact support
support@tech.alliancecan.ca

https://um-grex.github.io/docs/software/software-list/