University of Manitoba

Introduction to High Performance Computing software - Lmod modules

May 21, 2025.

Ali Kerrache

HPC Analyst

University of Manitoba

Outline

- ★ Software distribution on HPC clusters
- ★ How to find software packages and modules?
- ★ Software stacks on Grex
- ★ Hands on: using modules

University of Manitoba

Software distribution

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**)

Using a centralized HPC software stack:

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

Local installation: usually to $HOME {/home/$USER} or $PROJECT

- ★ **Get the code:** download the 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, …

★ Commercial Software: restricted [VASP, STATA, …]

➔ Contact support with some details about the license, …

➔ We install the program & protect it with a POSIX group.

Analysts / Sys admins

University of Manitoba

Software and modules

What are Modules?

- configuration files that contain instructions for modifying your software environment.
- The modular architecture allows multiple versions of the same application to be installed without conflict.
- contains instructions that modify or initialize environment variables such as PATH and LD_LIBRARY_PATH in order to use different installed programs.

Why modules?

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

University of Manitoba

Lmod and modules

★ 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/version
- module **help** <soft>/<version>
- module **purge**; module --force **purge**
- module **use** ~/modulefiles
- module **unuse** ~/modulefiles

```
[~@yak ]$  module list
_
Currently Loaded Modules:
SBEnv (S)

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

```
[~@login2 ~]$ module list
Currently Loaded Modules:
1) CCconfig                  4) gcc/12.3        (t)   7)
libfabric/1.18.0  10) openmpi/4.1.5   (m)  13) StdEnv/2023 (S)
2) gentoo/2023   (S)   5) hwloc/2.9.1       8) pmix/4.2.4      11)
flexiblas/3.3.1
3) gcccore/.12.3 (H)   6) ucx/1.14.1        9) ucc/1.2.0       12)
imkl/2023.2.0   (math)
```

University of Manitoba

HPC software stacks

- A set of compilers and libraries:
  - GCC, Intel compilers, …
  - Libraries: hdf5, boost, netcdf, …
- Modules hierarchy:
  - arch: branch for a given architecture {avx2; avx512}
  - CUDA: any program using GPU acceleration under the same tree.
  - Core modules: java, perl, … etc.
  - Compiler:
    - GCC: programs compiled with gcc
    - Intel: compiled with Intel
  - OpenMPI: Compiler/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, Intel}, MKL, PETSc, … etc.
  - Gaussian, ANSYS, MATLAB, … etc.
- ★ The Alliance (Compute Canada) environment [optional]: CCEnv
  - Switch to CCEnv; load a standard environment; choose the architecture[avx2/avx512], use module spider <soft>

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

For GPUs:

```
module load CCEnv
module load arch/avx2
module load StdEnv/2023
```

University of Manitoba

Software maintenance

- Compilers/Libraries and more:
  - Compilers: GCC [8.5 - 13.2]; Intel [2019, 2023], … etc.
  - Libraries: HDF5, PETSc, GSL, MKL, Libxc, Boost, ...
  - Gaussian, ANSYS, MATLAB, VASP, ORCA, MCR, Java, Python, R, … etc.
  - LAMMPS, GROMACS, 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 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

Lmod commands

★ **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/version
- module **help** <soft>/<version>
- module **purge;** module --force **purge**
- module **use** ~/modulefiles; module **unuse** ~/modulefiles

University of Manitoba

Hands on: locate modules

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

Exercise:

- ★ 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 many versions of the module.
    - If many, pick a version and run: module spider myprogram/version
    - Read the instructions and load 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.

University of Manitoba

Thank you for your attention

Any question?

University of Manitoba

Additional slides

module list: MC/Alliance

[~@narval2: ~]$ module list

Currently Loaded Modules:

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

- StdEnv/2023 loaded by default
- gcc/12.3 and openmpi/4.1.5
- Possibility to switch to: StdEnv/2020, …

module list: Grex

[~@yak ~]$ module list

Currently Loaded Modules:
1) SBEnv (S)

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

- SBEnv loaded by default
- No compiler loaded by default.
- Possibility to switch to:
  CCEnv and StdEnv/2023, …

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

Example of modules: python

```
[~@~]$  module spider python
```

```
[~@~]$ module spider python/3.9.6
[~@~]$ module load StdEnv/2020 python/3.9.6
```

```
[~@~]$ module spider python/3.12.4
[~@~]$ module load StdEnv/2023 python/3.12.4
```

Versions:

- - -

python/3.9.6

python/3.11.5

python/3.12.4

python/3.13.2

- - -

```
[~@~]$ module load python
[~@~]$ module list
1) CCconfig  4) gcc/12.3  (t) 7) libfabric/1.18.0    10) openmpi/4.1.5 (m)  13) StdEnv/2023   (S)
2) gentoo/2023 (S) 5) hwloc/2.9.1 8) pmix/4.2.4  11) flexiblas/3.3.1  14) python/3.11.5 (t)
3) gcccore/.12.3 (H)  6) ucx/1.14.1 9) ucc/1.2.0   12) blis/0.9.0
```

University of Manitoba

Example of modules: boost

```
[~@~]$ module spider boost
```

Versions:

- - -

boost/1.72.0
boost/1.76.0
boost/1.80.0
boost/1.82.0
boost/1.85.0

- - -

```
[~@~]$ module spider boost/1.82.0
```

StdEnv/2023 gcc/12.3
StdEnv/2023 intel/2023.2.1

```
[~@~]$ module load StdEnv/2023 intel/2023.2.1 boost/1.82.0
```

```
[~@~]$ module load StdEnv/2023 gcc/12.3 boost/1.82.0
```

```
[~@~]$ module load StdEnv/2023 gcc/12.3 boost/1.82.0
[~@~]$ module list
1) CCconfig  4) gcccore/.12.3 (H)    7) ucx/1.14.1          10) ucc/1.2.0                13) blis/0.9.0
2) gentoo/2023 (S)  5) gcc/12.3 (t)  8) libfabric/1.18.0  11) openmpi/4.1.5 (m)  14) boost/1.82.0 (t)
3) StdEnv/2023 (S)  6) hwloc/2.9.1   9) pmix/4.2.4         12) flexiblas/3.3.1
```

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/20.18.1                 rust/1.79.0
   adf/2021.106-impi                expect/5.45.4              julia/1.11.3        (D)    nodejs/22.4.1                  rust/1.86.0        (D)
   adf/2021.107-impi                fastqc/0.12.1              libaec/1.0.6               nodejs/22.5.1                  samtools/1.20
   adf/2023.104-impi                feko/2021.2                libvori/220621             nodejs/22.11.0          (D)    scons/3.1.2
   adf/2024.105-impi-aocl           ffmpeg/7.0.2               libxml2/2.11.9             nvtop/3.1.0                    singularity/4.1.2
   adf/2024.105-impi          (D)   fftw/3.3.10                matlab-proxy/0.24.2        openeye/2022.2.1               singularity/4.2.2  (D)
   admixture/1.3.0                  flex/2.6.4                 matlab/R2020B2             openjdk/11.0.22                skopeo/1.18.0
   ansys/21.1                       freeglut/3.4.0             matlab/R2022A              openjdk/17.0.11_9              snp-sites/2.5.1
   ansys/2023R2               (D)   gaussian/g16.b01           matlab/R2023B              openjdk/17.0.12_7              snpeff/5.2f
   ant/1.10.15                      gaussian/g16.c01      (D)  matlab/R2024A       (D)    openjdk/17.0.13_11             sparsehash/2.0.4
   arch/avx2                        gcc/8.5.0                  mcr/R2020b                 openjdk/21.0.2                 sqlite/3.35.5
   arch/avx512                (D)   git-annex/10.20250320      mcr/R2022a                 openjdk/21.0.3_9               stata/15.0-fagfs
   autotools/2022a                  git-lfs/3.6.1              mcr/R2023b                 openjdk/21.0.4_7               stata/18.0-ffin    (D)
   bamtools/2.5.2                   git/2.49.0                 mcr/R2024a          (D)    openjdk/21.0.5_11              structure/2.3.4
   beagle/5.4-20241029              globus/3.33.1              megahit/1.2.9              openjdk/21.0.6_7        (D)    subread/2.0.8
   birch/3.90                       gmp/6.2.1                  metashape-pro/2.1.3        openssl/3.4.0                  subversion/1.14.3
   buildah/1.39.3                   gmp/6.3.0             (D)  metashape-pro/2.2.0 (D)    ovito/3.12.0                   swig/4.2.0
   busco/5.8.3                      gnina/1.1                  micro/2.0.14               pandoc/3.6.1                   trimmomatic/0.39
   cfitsio/4.4.1                    gnina/2024            (D)  mii/1.1.1                  paraview-offscreen/5.10.1      unrar/7.10.2
   cfitsio/4.5.0              (D)   gnuplot/5.4.2              minimap2/2.28              picard/3.3.0                   vaspkit/1.5.1
   cmake/3.28.4                     gnuplot/6.0.2         (D)  molden/7.3                 podman-compose/1.3.0           velvet/1.2.10
   cmake/3.31.1               (D)   golang/1.24.2              mpfr/4.2.1                 podman-tui/1.5.0               vep/113.4
   code-server/4.99.1               haploview/4.2              mumax3/3.10                podman/5.4.2                   vim/9.1
   compleasm/0.2.6                  hmmer/3.4                  mummer/4.0.0rc1            prodigal/2.6.3                 visit/3.4.2
   cppzmq/4.10.0                    htgettoken/2.0-2           nbo/nbo7-2021              python/3.11.8                  voroplusplus/0.4.6
   cuda/11.8.0                      intel-one/2023.2           nextflow/24.10.0           qiime2/2024.10                 wine/10.0
   cuda/12.2.2                      intel-one/2024.1      (D)  ninja/1.10.2               quast/5.3.0                    xz/5.6.4
   cuda/12.4.1                (D)   intel/2023.2               nodejs/18.20.2             r/4.3.3                        yaml-cpp/0.8.0
   cudnn/8.8.1.3+cuda-11.8.0        intelmpi/2019.8            nodejs/18.20.4             ramalama/0.7.3                 z3/4.13.4
   deepvariant/1.8.0-gpu            intelmpi/2021.10      (D)  nodejs/18.20.5             ratarmount/1.0.0               zstd/1.5.6
   deepvariant/1.8.0          (D)   jellyfish/2.3.1            nodejs/20.12.2             rclone/1.67.0
   dejagnu/1.6.3                    jemalloc/5.3.0             nodejs/20.15.1             rclone/1.68.2           (D)
   diamond/2.1.10                   jq/1.7                     nodejs/20.16.0             rs-server/2024.12.1-563

------------------------------------------------------------------ /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.

  Where:
   S:  Module is Sticky, requires --force to unload or purge
   L:  Module is loaded
   D:  Default Module
   E:  Extension that is provided by another module
```

module avail

module spider python
module spider java

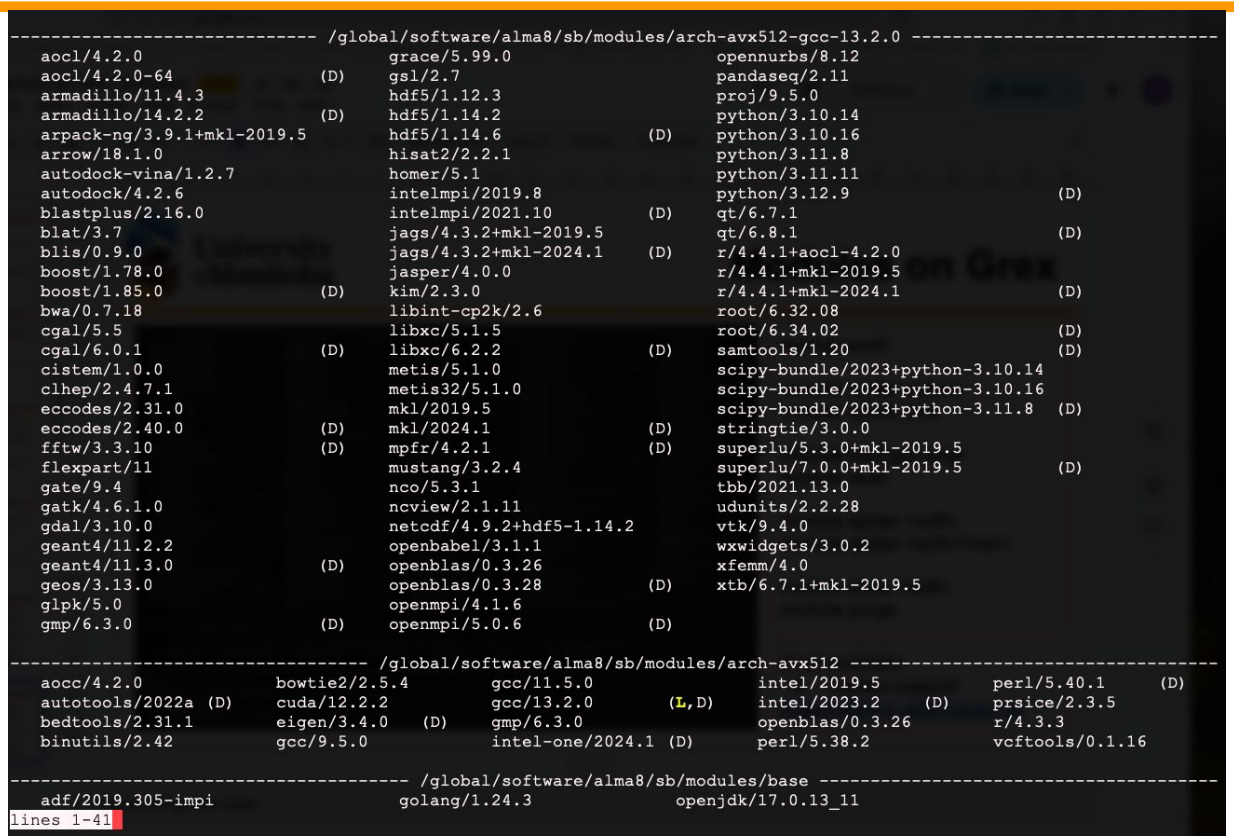
module load gcc ompi
module avail

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

module show <soft>
module purge

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

Modules on Grex



module load arch/avx512 gcc
module avail

- Shows a long list of all modules installed with gcc under arch/avx512

module load arch/avx512 gcc
module load openmpi
module avail

- It will shows a long list of all modules installed with gcc and openmpi under arch/avx512

module spider <software name>

Cvmfs on Grex

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

[~@yak ~]$  ls -1 /cvmfs/
cvmfs-config.computecanada.ca
oasis.opensciencegrid.org
restricted.computecanada.ca
singularity.opensciencegrid.org
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
```

Find and load Gaussian

Gaussian: restricted software; requires a registration

https://docs.alliancecan.ca/wiki/Gaussian

https://um-grex.github.io/grex-docs/docs/grex/software/specific/gaussian/

```
[~@yak ]$  module spider gaussian
---------------------------------------------------------------------------------------------------------------------------------------------------------
  gaussian:
---------------------------------------------------------------------------------------------------------------------------------------------------------
     Versions:
        gaussian/g16.b01
        gaussian/g16.c01
```

```
[~@yak ~]$  module load gaussian/g16.c01
    Loading  Gaussian version 16.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
---------------------------------------------------------------------------------------------------------------------------------------------------------
```

Available: Grex, Cedar and Graham

Find and load ORCA

ORCA:

https://docs.alliancecan.ca/wiki/ORCA

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

- **restricted software**
- **requires a registration**

[~@yak]$ module spider orca

[~@yak]$ module spider **orca/6.0.1**

---

orca:

Versions:
orca/5.0.4
orca/6.0.1

[~@~]$ module load arch/avx512 gcc/13.2.0 openmpi/4.1.6 **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

Available: Grex and all Alliance clusters

Find and load LAMMPS

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

```
[~@~]$  module spider lammps/2024-08-29p1
```

```
[~@yak ]$  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
```

Find and load QE

```
[~@yak ]$  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
----------------------------------------------------------------------------
  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
----------------------------------------------------------------------------
```

```
[~@yak ]$  module spider espresso/7.4.1
```

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

University of Manitoba

MATLAB and MCR

```
[~@yak ~]$  module spider matlab
----------------------------------------------------------------------------------------------------------------------
  uofm/matlab:
----------------------------------------------------------------------------------------------------------------------
     Versions:
        matlab/R2020B2
        matlab/R2022A
        matlab/R2023B
        matlab/R2024A
     Other possible modules matches:
        matlab-proxy
----------------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "uofm/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
----------------------------------------------------------------------------------------------------------------------
```

```
[~@yak ~]$  module spider mcr
Versions:
        mcr/R2020b
        mcr/R2022a
        mcr/R2023b
        mcr/R2024a
```

University of Manitoba

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: {separate talk}
  - build the image and run your program using the container

University of Manitoba

Local installation

- R packages: minimal installation
  - R as modules: users can install the packages in their home directory.
- Python as modules: python and scipy-stack
  - users can install the packages needed in their home directory.
- Perl as 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

Local installation: R packages

R packages: rgdal, adegenet, stats, rjags, dplyr, … etc.

Choose a module version: module spider r

Load R and dependencies (gdal, geos, jags, gsl, udunits… etc):

```
module load gcc r gdal udunits
```

Launch R and install the packages:

```
~$ R
> install.packages(”sp")
'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(”dplyr")
```

University of Manitoba

Local installation: Python

- ★ Load the modules:
  - module load python
- ★ 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/3.1
virtualenv ~/my_venv
source ~/my_venv/bin/activate
pip install cutadapt
deactivate
```

```
module load gcc python/3.11.2
source ~/my_venv/bin/activate
cutadapt [+options]
deactivate
```

https://docs.alliancecan.ca/wiki/Python

University of Manitoba

Local installation: Perl

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
```

University of Manitoba

Java applications

- ★ Download and unpack the code
- ★ Load java module: module load java
- ★ Run the code

- ★ Example: Trimmomatic
  - wget http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/Trimmomatic-0.39.zip
  - unzip Trimmomatic-0.39.zip

- ★ Run the code

```
module load java
java -jar <path to>/trimmomatic-0.39.jar {+options if any}
```

University of Manitoba

Installation with make: 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

Installation with configure/make

- ★ Download and unpack the code: wget, … gunzip, … etc.
- ★ Load the modules and dependencies: module load gcc ompi 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

University of Manitoba

Installing: 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

Example: 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

Example: 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

Example with cmake/make

- ★ Download and unpack the code: wget, … gunzip, … etc.
- ★ Load the modules and dependencies: module load gcc ompi fftw
- ★ 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 {+other options}
- ★ Compile and test:
  - make; make -j8
  - make check; make test
- ★ Install the program:
  - make install

University of Manitoba

Cmake options for 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 about HPC software

- 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

Summary about HPC workflow

- Account and active role:
  - CCDB
- Have a look to the documentation:
  - Hardware, available tools, …
  - policies?
  - login nodes
  - storage, …
- Tools to connect and transfer files
- Access to storage: home, scratch, project
- Access to a program to use:
  - Install the program or ask for it.
  - Use the existing modules

- Test jobs:
  - Login node
  - Interactive job via salloc
- Write a job script:
  - Slurm directives
  - Modules
  - Command line to run the code
- Monitor jobs:
  - Sacct; seff, optimize jobs
- Analyze data:
  - Post processing
  - Visualization

University of Manitoba

Software layers in HPC

User layer: Python packages, Perl and R modules, home made codes, …

Software stacks: modules for Intel, PGI, OpenMPI, CUDA, MKL, high-level applications. Multiple architectures (sse3, avx, avx2, avx512)

Nix or gentoo: GNU libc, autotools, make, bash, cat, ls, awk, grep, etc.

Gray area: Slurm, Lustre client libraries, IB/OmniPath/InfiniPath client libraries (all dependencies of OpenMPI) in Nix {or gentoo} layer, but can be overridden using PATH & LD_LIBRARY_PATH.

OS: kernel, drivers, daemons, anything privileged (e.g. the sudo command): always local. Some legally restricted software too (VASP).

User

Analysts

Sys. Admin