HPC Software – Modules, Libraries & Software

JSC Training Course

November 22, 2022 | R. Schöbel | JSC

Member of the Helmholtz Association

JÜLICH Forschungszentrum

Outline

1 Modular Programming

2 Preinstalled HPC Software Packages

- Navigating Modules
- Mathematical Libraries & Application Software

3 Containers

4 Further Information

Modular Programming

Software Implementation & Libraries

A **library** is a collection of resources.
In computer science: configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values or type specifications

- **Don’t reinvent the wheel**: Recoding is time-consuming and error-prone
- **Best practice for own code**: Design interfaces and do different implementation separately (ideally separate files)

Modular Programming

Compiling & Linking

Example:

- Code decomposition into an executable and an outsourced subprogram *(C++: Link object files for different modules)*

  ```
  g++ -g -c -o main.o main.cpp
  g++ -g -c -o alibrary.o alibrary.cpp
  g++ -g main.o alibrary.o -o main
  ```

  (In real life: Use **make** or **cmake**)
- Copy *.o* and *.h* to separate directories (e.g. */base/lib/alibrary.o*) and add
  - The **-L** option for the path to the library's object, the object code using **-l**

    ```
    g++ -g -L/base/lib -o main main.o -lalibrary
    ```

  - The **-I** flag for include directories

Modular Programming

Compiling & Linking

- For libraries that are not in standard directories, you need to set **-I**/base/include and **-L**/base/lib for compiling/linking
- Or define linux **environment variables**:
  - You either enter the *export*-commands on the linux prompt before compiling, or, copy them into to the.bashrc file in your home folder
- Usually you do not need **-I** or **-L** for libraries accessed using the *module load* command on our supercomputers

Modular Programming

Installing Libraries from Source

- Library is not in the software module stack
- Common installation procedure:
  - make
    ```
    ./configure –prefix=base
    make; make install
    ```
  - cmake
    ```
    cmake -DCMAKE_INSTALL_PREFIX=base
    make; make install
    ```
  - Choose *base* as a directory you have write permissions

Preinstalled HPC Software Packages

EasyBuild

Preinstalled HPC Software Packages

Navigating Modules

- Tools are available through **"modules"**
  - Allows to easily manage different versions of programs
  - Works by dynamic modification of a user's environment
- Module setup based on **EasyBuild** and **lmod**
  - **"Staged"**, hierarchical setup
  - Automatically manages dependencies via **"toolchains"**
- Consistent setup on JURECA, JUSUF and JUWELS (Cluster & Booster)

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

- **Base:** GCCcore
- **Compiler:**
  - Intel compiler
  - GNU compiler
  - NVIDIA (CUDA) compiler
- **MPI libraries:**
  - ParastationMPI
  - Intel MPI
  - OpenMPI
- **Math libraries:** e.g. MKL

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

Current software stage is 2022, stage 2023 is under construction and can already be used

- **Base:** GCCcore *(11.2.0)*
- **Compiler:**
  - Intel compiler *(Intel 2021.4.0)*
  - GNU compiler *(GCC 11.2.0)*
  - NVIDIA (CUDA) compiler *(NVHPC TBD)*
- **MPI libraries:**
  - ParastationMPI *(ParaStationMPI 5.5.X)*
  - Intel MPI *(Intel MPI 2021.X.Y)*
  - OpenMPI *(OpenMPI 4.1.X)*
- **Math libraries:** e.g. MKL *(2021.4.0)*

Preinstalled HPC Software Packages

Modules Environment

Available Compiler/MPI Combinations in Stages/2022

Compiler	MPI	Cuda available
GCC *11.2.0*	ParaStationMPI *5.5.X*	yes
GCC *11.2.0*	OpenMPI *4.1.X*	yes
NVHPC *TBD*	ParaStationMPI *5.5.X*	yes
NVHPC *TBD*	OpenMPI *4.1.X*	yes
Intel *2021.4.0*	ParaStationMPI *5.5.X*	yes
Intel *2021.4.0*	IntelMPI *2021.X.Y*	no
Intel *2021.4.0*	OpenMPI *4.1.X*	yes

Preinstalled HPC Software Packages

Modules Environment

Available Combinations of Compilers/MPI/Math libraries in Stages/2022

Compiler	MPI	Math	Cuda	arch
GCC	ParaStationMPI	MKL	yes	Intel&AMD
GCC	OpenMPI	MKL	yes	Intel&AMD
Intel	ParaStationMPI	MKL	yes	Intel&AMD
Intel	IntelMPI	MKL	no	Intel
Intel	OpenMPI	MKL	yes	Intel&AMD
NVHPC	ParaStationMPI	MKL	yes	GPU

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

The most important module command

- module load <*name*> or ml <*name*>

- **GCCcore** is preloaded, which enables a lot of base software
- For HPC software you have to load a **compiler**, to expand the module tree
  e.g. `ml GCC` *(Default is 11.2.0)*
- Then you load an **MPI** version
  e.g. `ml ParaStationMPI` *(Default is 5.5.X)*
- Then you can load any other **math or application package**
  e.g. `ml PETSc/3.16.3`

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

The most important module command

- module load <*name*> or ml <*name*>

Hierarchical modules

- **GCCcore** is preloaded, which enables a lot of base software
- For HPC software you have to load a **compiler**, to expand the module tree
  e.g. `ml GCC` *(Default is 11.2.0)*
- Then you load an **MPI** version
  e.g. `ml ParaStationMPI` *(Default is 5.5.X)*
- Then you can load any other **math or application package**
  e.g. `ml PETSc/3.16.3`

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

The most important module command

- module load <*name*> or ml <*name*>

Hierarchical modules

- **GCCcore** is preloaded, which enables a lot of base software
- For HPC software you have to load a **compiler**, to expand the module tree
  e.g. `ml GCC` *(Default is 11.2.0)*
- Then you load an **MPI** version
  e.g. `ml ParaStationMPI` *(Default is 5.5.X)*
- Then you can load any other **math or application package**
  e.g. `ml PETSc/3.16.3`

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

The most important module command

- module load <*name*> or ml <*name*>

Hierarchical modules

- **GCCcore** is preloaded, which enables a lot of base software
- For HPC software you have to load a **compiler**, to expand the module tree
  e.g. `ml GCC` *(Default is 11.2.0)*
- Then you load an **MPI** version
  e.g. `ml ParaStationMPI` *(Default is 5.5.X)*
- Then you can load any other **math or application package**
  e.g. `ml PETSc/3.16.3`

Preinstalled HPC Software Packages

Toolchains: Underlying Compiler and MPI Libraries

The most important module command

- module load <*name*> or ml <*name*>

Hierarchical modules

- **GCCcore** is preloaded, which enables a lot of base software
- For HPC software you have to load a **compiler**, to expand the module tree
  e.g. `ml GCC` *(Default is 11.2.0)*
- Then you load an **MPI** version
  e.g. `ml ParaStationMPI` *(Default is 5.5.X)*
- Then you can load any other **math or application package**
  e.g. `ml PETSc/3.16.3`

Preinstalled HPC Software Packages

Modules Environment

- After loading compiler and MPI `ml avail` shows the software available with that combination
- `ml avail name` and `ml help name` will show you details about the *name* package
- Many libraries are available for more than one combination/toolchain

- Write e-mail to `sc@fz-juelich.de` if you want special versions or new software
  - No guarantee the software will be installed
- $EBROOT*NAME* is the root directory where the library is installed

Preinstalled HPC Software Packages

Modules Environment

- After loading compiler and MPI `ml avail` shows the software available with that combination
- `ml avail` *name* and `ml help` *name* will show you details about the *name* package
- Many libraries are available for more than one combination/toolchain
- Write e-mail to `sc@fz-juelich.de` if you want special versions or new software
  - No guarantee the software will be installed
- $EBROOT*NAME* is the root directory where the library is installed

Preinstalled HPC Software Packages

Modules Environment

- After loading compiler and MPI `ml avail` shows the software available with that combination
- `ml avail` *name* and `ml help` *name* will show you details about the *name* package
- Many libraries are available for more than one combination/toolchain
- Write e-mail to `sc@fz-juelich.de` if you want special versions or new software
    - No guarantee the software will be installed
- $EBROOT*NAME* is the root directory where the library is installed

Preinstalled HPC Software Packages

Modules Environment

- `ml spider` *name* shows whether a library is available in the current stage and in which versions

  e.g. module spider petsc

  ⇒ PETSc/3.14,
  PETSc/3.16.3, ...

- `ml spider` *name/version* shows which environment you have to load before you can load that version

  e.g. module spider PETSc/3.16.3

  ⇒ Stages/2022 + GCC/11.2.0 + OpenMPI/4.1.2,
  Stages/2022 + GCC/11.2.0 + ParaStationMPI/5.5.0-1, ...

- Some packages are hidden. To see them use `ml -show-hidden spider` *name*

Preinstalled HPC Software Packages

Modules Environment

- `ml spider` *name* shows whether a library is available in the current stage and in which versions

  e.g. module spider petsc

  ⇒ PETSc/3.14,
  PETSc/3.16.3, ...

- `ml spider` *name/version* shows which environment you have to load before you can load that version

  e.g. module spider PETSc/3.16.3

  ⇒ Stages/2022 + GCC/11.2.0 + OpenMPI/4.1.2,
  Stages/2022 + GCC/11.2.0 + ParaStationMPI/5.5.0-1, ...

- Some packages are hidden. To see them use `ml -show-hidden spider` *name*

Preinstalled HPC Software Packages

Modules Environment

- `ml spider` *name* shows whether a library is available in the current stage and in which versions

  e.g. module spider petsc

  ⇒ PETSc/3.14,
  PETSc/3.16.3, ...

- `ml spider` *name/version* shows which environment you have to load before you can load that version

  e.g. module spider PETSc/3.16.3

  ⇒ Stages/2022 + GCC/11.2.0 + OpenMPI/4.1.2,
  Stages/2022 + GCC/11.2.0 + ParaStationMPI/5.5.0-1, ...

- Some packages are hidden. To see them use `ml -show-hidden spider` *name*

Preinstalled HPC Software Packages

Modules Environment

- For R, Python and Perl we use bundles
  - You might be looking for a software package that is part of a bigger module
- Use `ml key software`
  `ml key numpy` will suggest `SciPy-Stack`
- You can use then `ml spider` to find out how to load the module

Preinstalled HPC Software Packages

Modules Environment

Stages

- The whole software stack of JURECA, JUSUF, JUWELS Cluster and Booster will be updated regularly
- Current stage is 2022 (*2023 is under construction*)
- Old stages are still accessible on JUWELS Cluster, but no guarantee
- To check availability in other stages first type

```
ml use $OTHERSTAGES
```

Preinstalled HPC Software Packages

Mathematical Libraries: MKL

Contents of Intel MKL

- BLAS, Sparse BLAS, CBLAS
- LAPACK, ScaLAPACK
- Iterative Sparse Solvers, Trust Region Solver
- Vector Math Library
- Vector Statistical Library
- Fourier Transform Functions
- Trigonometric Transform Functions
- GMP routines, Poisson Library, ...

Preinstalled HPC software packages

Mathematical Libraries: Sequential Packages

Public domain Libraries

- LAPACK (Linear Algebra PACKage)
- ARPACK (ARnoldi PACKage)
- GSL (Gnu Scientific Library)
- GMP (Gnu Multiple Precision Arithmetic Library)
- METIS (Serial Graph Partitioning and Fill-reducing Matrix Ordering)
- SCOTCH (Graph Partitioning)

Commercial library

NAG Fortran and C Library: on JURECA only

Preinstalled HPC Software Packages

Mathematical Libraries: Parallel Packages

- ScaLAPACK (Scalable Linear Algebra PACKage)
- ELPA (Eigenvalue SoLvers for Petaflop-Applications)
- FFTW (Fastest Fourier Transform of the West)
- MUMPS (MUltifrontal Massively Parallel sparse direct Solver)
- ParMETIS (Parallel Graph Partitioning)
- PT-SCOTCH (Parallel Graph Partitioning)
- Hypre (high performance preconditioners)
- PARPACK (Parallel ARPACK), Eigensolver
- SPRNG (Scalable Parallel Random Number Generator)
- SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers)

GPU Library

- MAGMA, Matrix Algebra on GPU and Multicore Architectures

Parallel Systems

- PETSc, toolkit for partial differential equations
  - PETSc for Python (petsc4py)
- SLEPc Scalable Library for Eigenvalue Problem Computations Extension to PETSc for the computation of eigenvalues and eigenvectors

Software for Materials Science

Package	JURECA	JUWELS
Abinit	yes	yes
Amber	no	yes
CP2K	yes	yes
CPMD	yes	no
GPAW	yes	yes
Gromacs	yes	yes
LAMMPS	yes	yes
NAMD	yes	yes
NWChem	yes	no
QuantumEspresso	yes	yes

Software for Computational Engineering

- JURECA Cluster and JUWELS
- CFD Package **OpenFOAM**
- Commercial **FEM Software**
  - **ANSYS**, **LS-DYNA** , **COMSOL** are technically maintained on JURECA Cluster only
  - **Licenses** must be provided by the **User!**

Containers

Apptainer

Containers

What they provide

- Containers package up pieces of software in a way that is **portable and reproducible**, they ...
  - manage different versions of programs
  - are more lightweight than virtual machines
  - provide the ability to build, ship, and run applications
- Some examples are Docker, Shifter, and **Apptainer/Singularity**
- They typically use so-called **"images"**
  - contain a file system including a minimal operating-system, the application, and some metadata

Apptainer Containers

First steps

- We provide an up-to-date version of **Apptainer**
  - Formerly, we provided Singularity on the Systems
  - We have replaced Singularity by Apptainer, a fork maintained by the linux foundation
- To be granted access to the container runtime, you have to go to our user portal **JuDoor**
  - *Software*
  - *Request access to restricted software*
  - *Access to other restricted software*
  - *Container*
  - *Get Access*
  - *Accept the Service Level Description*

Apptainer Containers

First steps

Environment variables:

- It might be helpful to overwrite some Apptainer environment variables

```
export APPTAINER_CACHEDIR =$(mktemp -d -p <DIR>)
```

```
export APPTAINER_TMPDIR= $(mktemp -d -p <DIR>)
```

Download an image:

- Use the pull command to download pre-built images from an external resource like Docker Hub

```
apptainer pull centos.sif docker://centos:7
```

Apptainer Containers

First steps

Call an executable:

- The shell command allows you to spawn a new shell within your container and interact with it

```
srun -N1 -p <part> –gres gpu:1 –pty apptainer shell –nv centos.sif
```

- To Slurm, Singularity is just another executable and can be called as such

Container Build System:

- JSC provides a build system that can build images on behalf of the user, based on a Docker- or Singularity-file
- For further information see `https://apps.fz-juelich.de/jsc/hps/jureca/container-runtime.html`

Further information and JSC-people

http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/_node.html
http://www.fz-juelich.de/ias/jsc/jureca
http://www.fz-juelich.de/ias/jsc/juwels
http://www.fz-juelich.de/ias/jsc/jusuf

Mailto
R. Schöbel: Parallel mathematical Libraries
r.schoebel@fz-juelich.de

Supercomputer support:
sc@fz-juelich.de