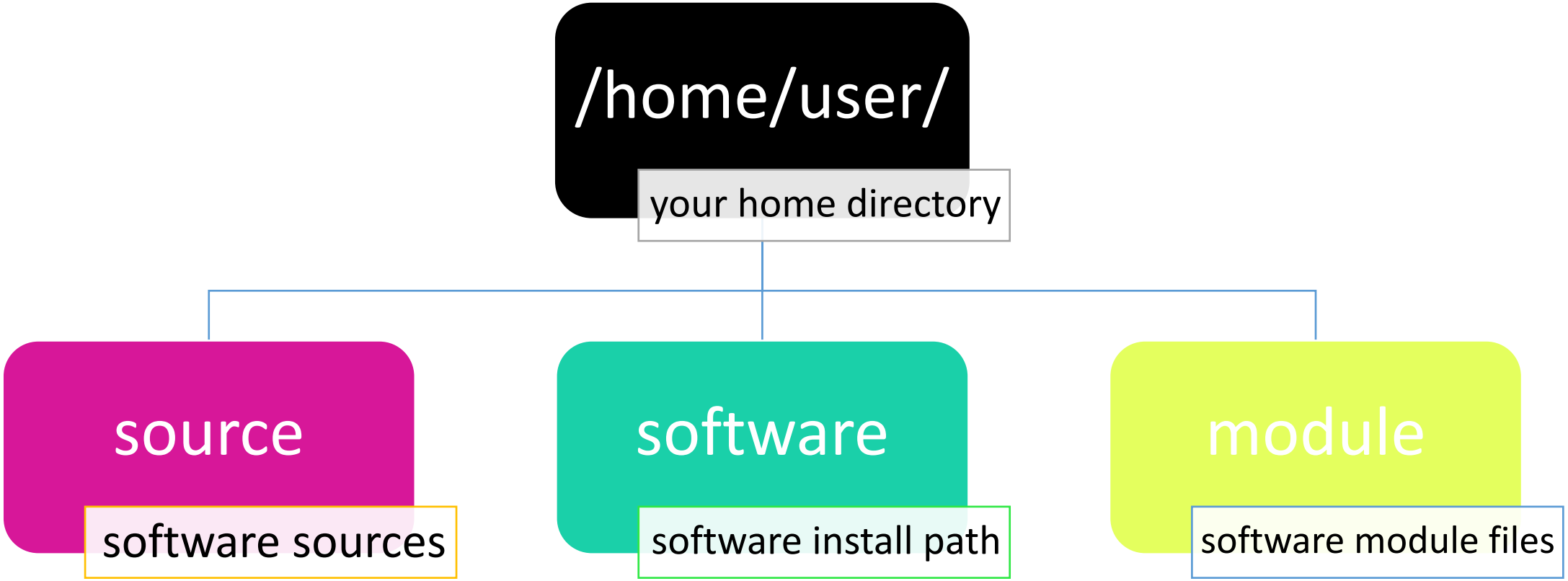
HPC series

install HPC software

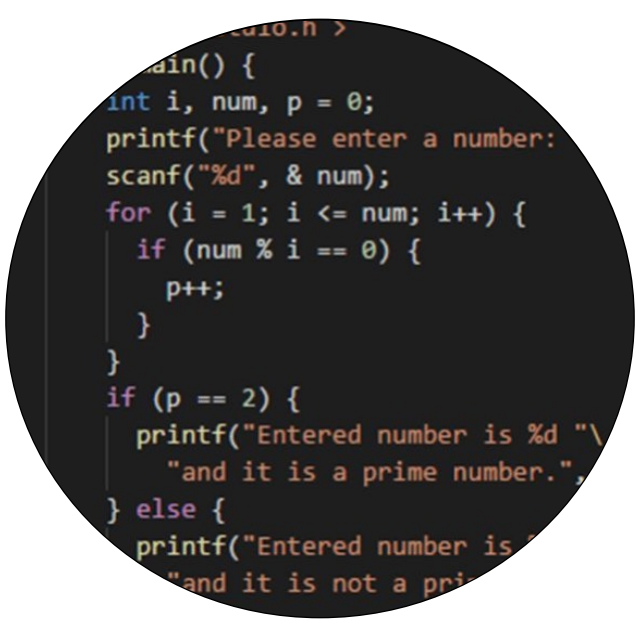
Author:
Sami Ait Ali Oulahcen

Rabat, Morocco
15 Aug 2023

Organize your workspace



Install your own software



Via package managers:

easy & mostly reproducible

auto manages dependencies

adds a layer of abstraction

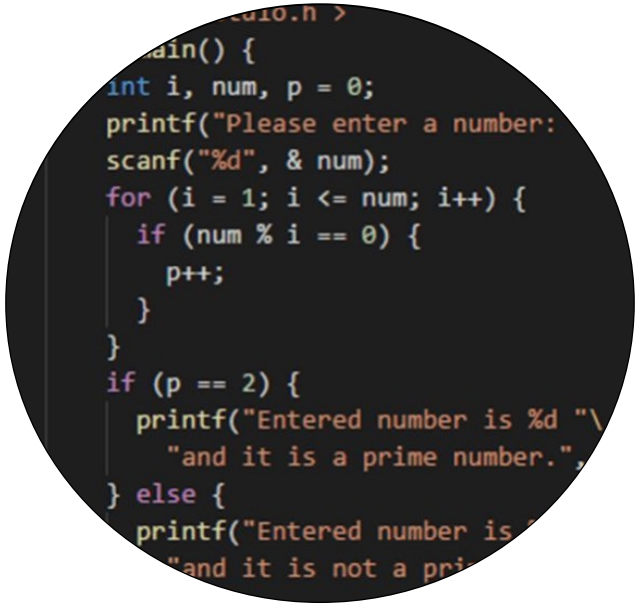
From source code:

full control

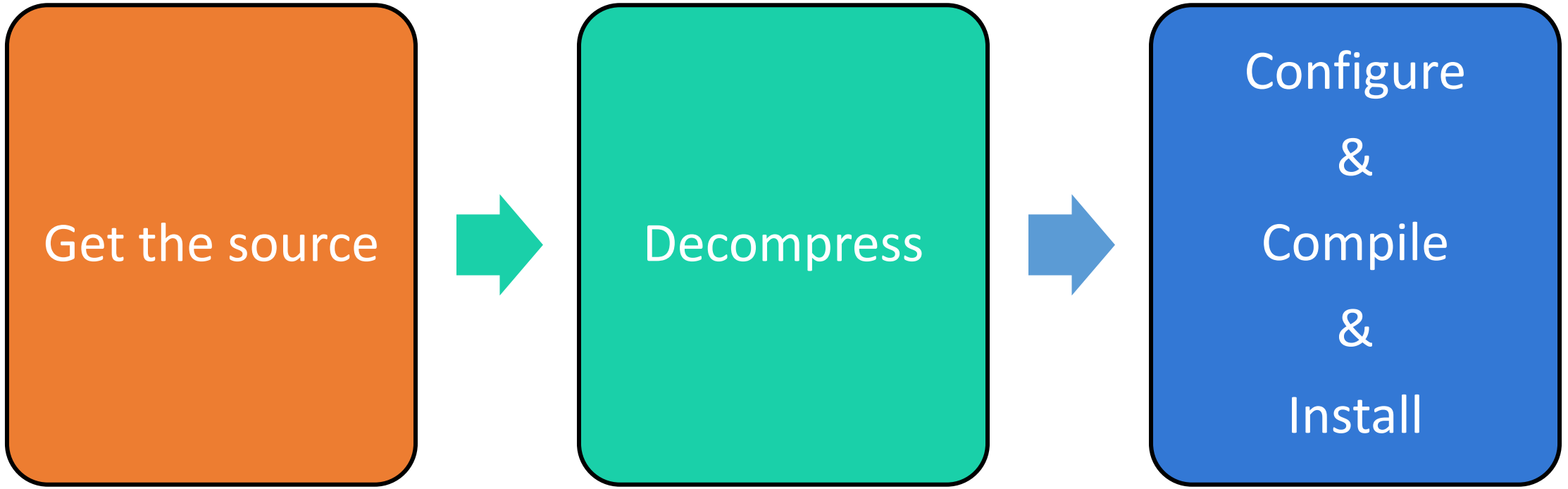
highly custumizable

can take time & effort

Install from source code



Install software from source code



Install software from source code

Download software source code to /home/user/source :

```
cd ~/source
wget <put-link-here>
OR
git clone <put-git-link-here>
```

Install software from source code

There are numerous types of compression, here are the most commonly used:

```
tar -xf file.tar
tar -xzf file.tar.gz
tar -xjf file.tar.bz2
```

```
tar -xJf file.tar.xz
tar --lzip -xf file.tar.lz
unzip file.zip
```

Install software from source code


```
./configure --prefix=/home/user/software/name/version
OR
cmake -DCMAKE_INSTALL_PREFIX=/home/user/software/name/version
make
make install
```

Create module files for software

`vi ~/module/<sofware-name>`

```
#%Module1.0
set base /home/user/software/name/version
prepend-path PATH            $base/bin
prepend-path LD_LIBRARY_PATH $base/lib
```

Using your modules

You can load your modules using their full path:

```
module load /home/USER/module/software
```

Or create a .modulerc file that includes your MODULEPATH:

```
vi ~/.modulerc
```

```
#%Module1.0
module use --append /home/user/module
```

```
source ~/.modulerc
```

Practice

Get into pairs

Choose a bioinformatics software and install it from source code

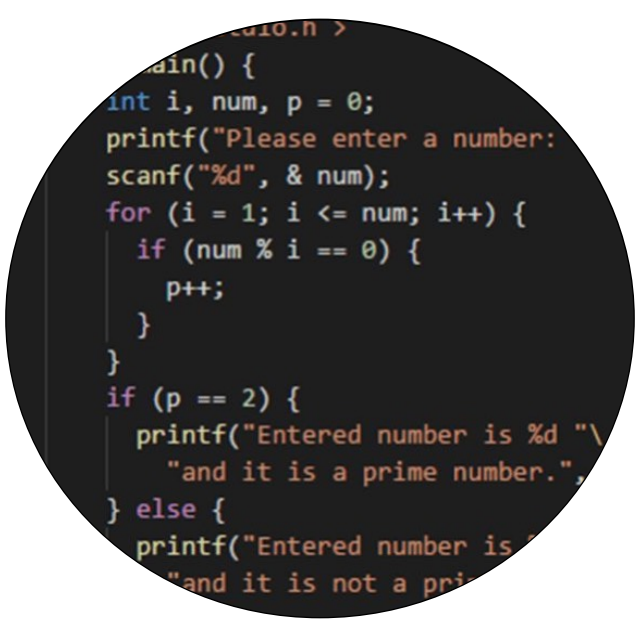
If needed, ask administrator to install dependencies

Build module file for software

Load the software module & run an example simulation

Examples: BLAST, LAMMPS, AMPHORA, ARB, GROMACS, SPAdes

Install your own software



Via package managers:

easy & mostly reproducible

auto manages dependencies

adds a layer of abstraction

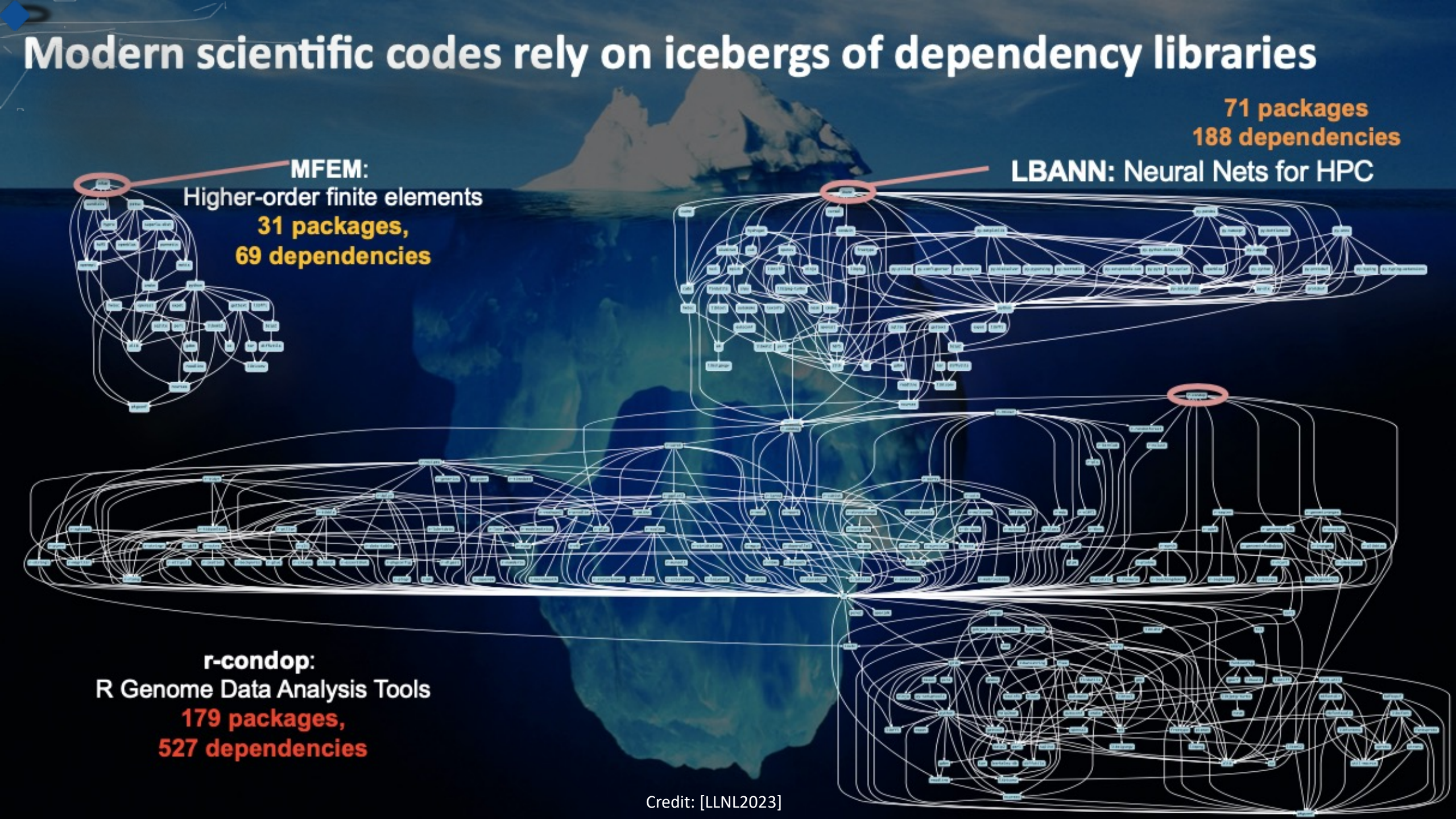
From source code:

full control

highly custumizable

can take time & effort

Modern scientific codes rely on icebergs of dependency libraries



Credit: [LLNL2023]

Install via package managers

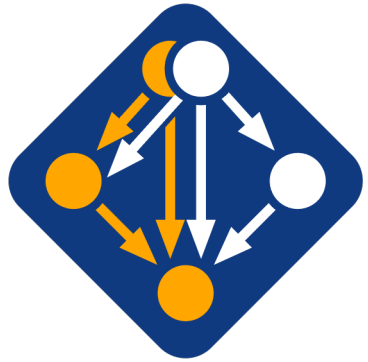
spack

From spack.io:

“Spack is a package manager for supercomputers, Linux, and macOS. It makes installing scientific software easy. Spack isn’t tied to a particular language; you can build a software stack in Python or R, link to libraries written in C, C++, or Fortran, and easily swap compilers or target specific microarchitectures.”

Approx. 10000 software available

Possiblity to create virtual environments



Using spack

Look for a software:

```
spack list *soft*
```

Show all versions:

```
spack versions soft
```

Install:

```
spack install soft @version
```

Specify dependencies and compilers:

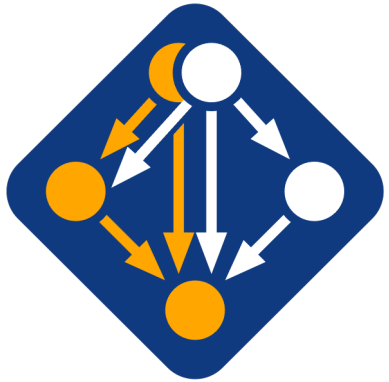
```
spack install soft ^dependency @version %compiler
```

Show installed software:

```
spack find soft
```

Uninstall:

```
spack uninstall soft%compiler@version
```



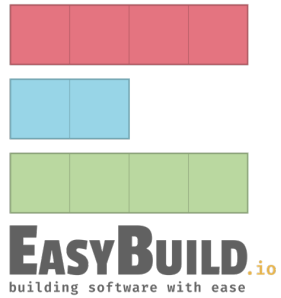
EasyBuild

From easybuild.io

"EasyBuild is a software build and installation framework that allows you to manage (scientific) software on High Performance Computing (HPC) systems in an efficient way."

Supports 3161 software packages (incl. toolchains, bundles)

Automatically creates module files



Using EasyBuild

Look for a software:

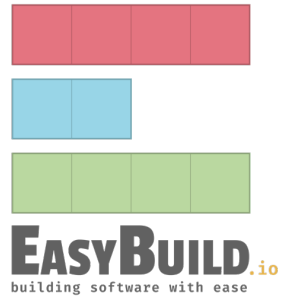
```
eb -S soft
```

Install:

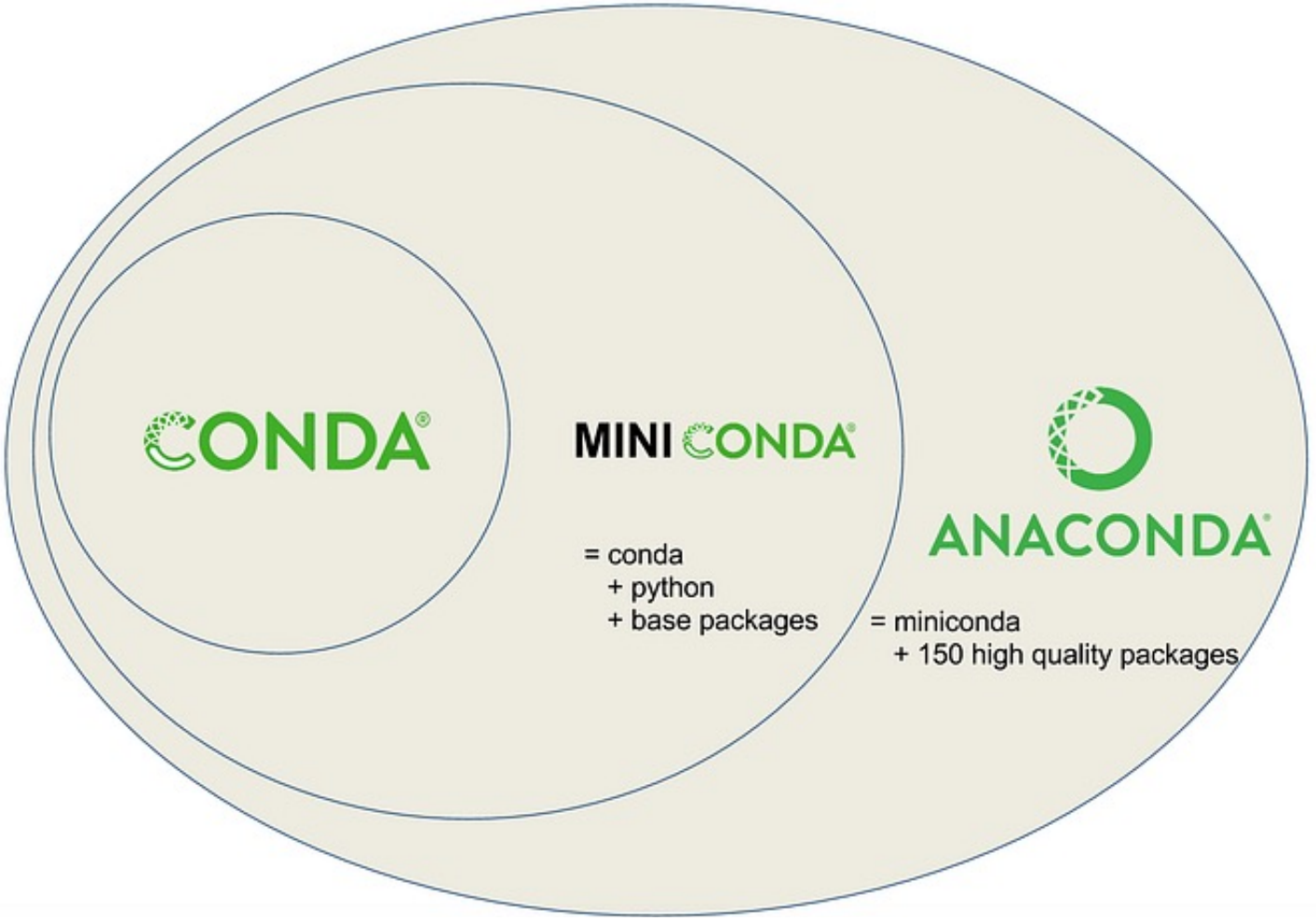
```
eb soft-ver.eb -r --sourcepath=~/software/
```

To use installed software, load the module:

```
module use /home/$USER/software/Easybuild/all
module load software
```



Anaconda



Credit: [PLNM2020]

Create a conda virtual environment

Create environment directory, then declare it as CONDA_ENVS_PATH:

```
mkdir /home/$USER/environment
echo ”export CONDA_ENVS_PATH=/home/$USER/environment” >> ~/.bashrc
source ~/.bashrc
```

Load Anaconda module:

```
module load Anaconda3/2023.03-1
```

Create virtual environment:

```
conda create -n lab1
```

Activate environment:

```
source activate lab1
```

Install software using conda

Many channels available. Channel priorities can be set as config or with install command:

```
conda config --add channels conda-forge
conda config --add channels bioconda
conda config --add channels defaults
conda config --set channel_priority strict
```

Install software:

```
conda install soft=version=build
```

When you're done, deactivate environment:

```
source deactivate
```

Practice

Create a conda virtual environment named condalab0

Activate condalab0

Install your favourite bioinformatics software in it

Deactivate condalab0

Write an sbatch script that uses condalab0 to run an example simulation

References

Web references:

- [WIKI2023] Frontier_Supercomputer retrieved from https://wikipedia.org on 17.07.2023
- [LLNL2023] LLNL-PRES-806064 retrieved from https://spack-tutorial.readthedocs.io on 08.08.2023
- [PLNM2020] conda-vs-miniconda-vs-anaconda by Planemo retrieved from https://towardsdatascience.com/managing-project-specific-environments-with-conda-b8b50aa8be0e on 04.08.2023
- [MRWN2022] HPC samples by MARWAN HPC team available at https://github.com/HPC-MARWAN-Team/hpc_samples/tree/master/Anaconda consulted on 09.08.2023