Leibniz Supercomputing Centre

of the Bavarian Academy of Sciences and Humanities

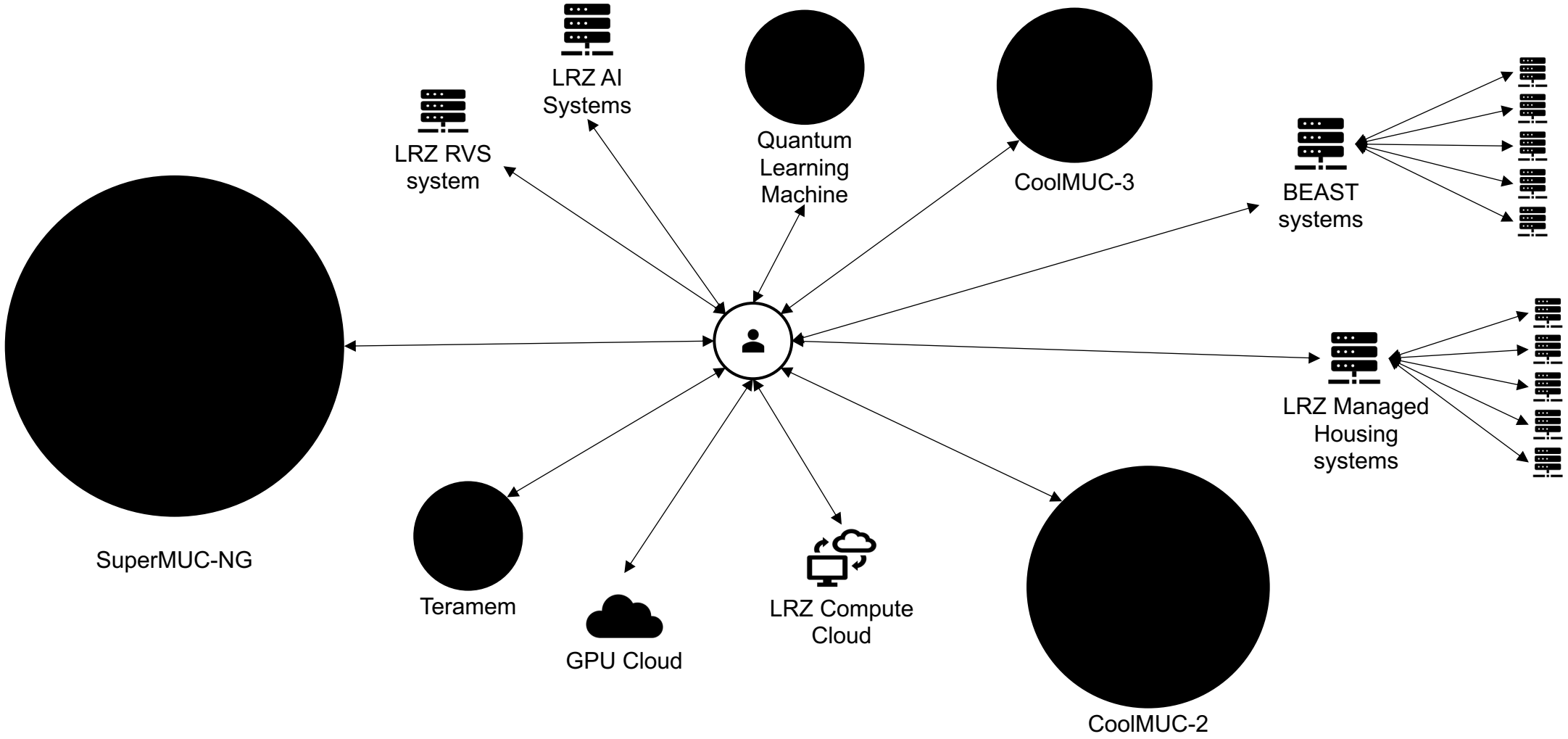
Leibniz Supercomputing Centre

Software Ecosystem at LRZ | 14.10.2022 | Nisarg Patel

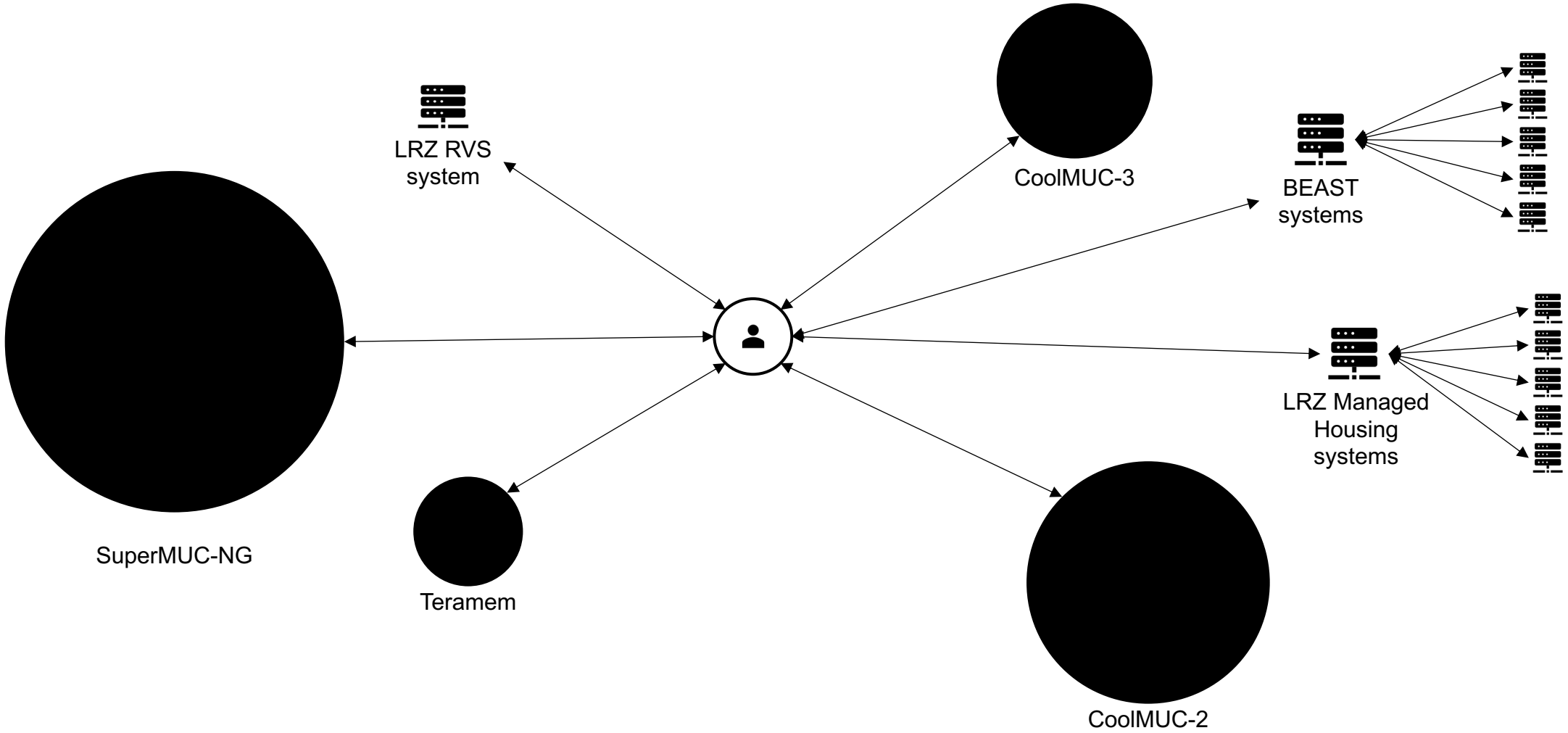
Motivation

- Introducing compute resources and software ecosystem on it.
- Understating the complexity in providing a fully-featured software – “dependency-hell”.
- LRZ software stack overview - list of modules available for users
- Using Spack on top of LRZ provided software ecosystem
  - Chaining & User Spack
  - Spack environments
- User Spack internals and in-dept view

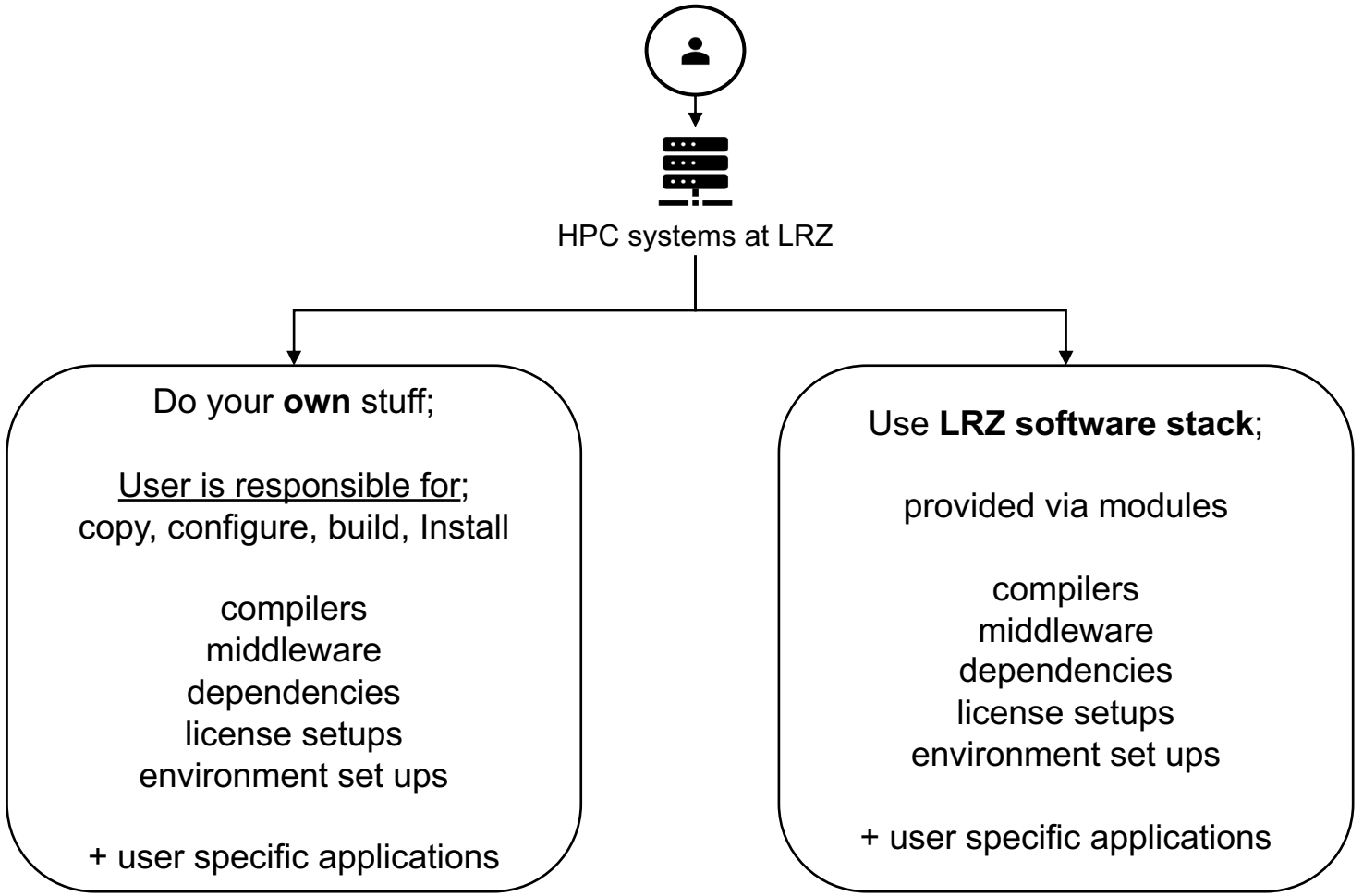
Available compute resources at LRZ



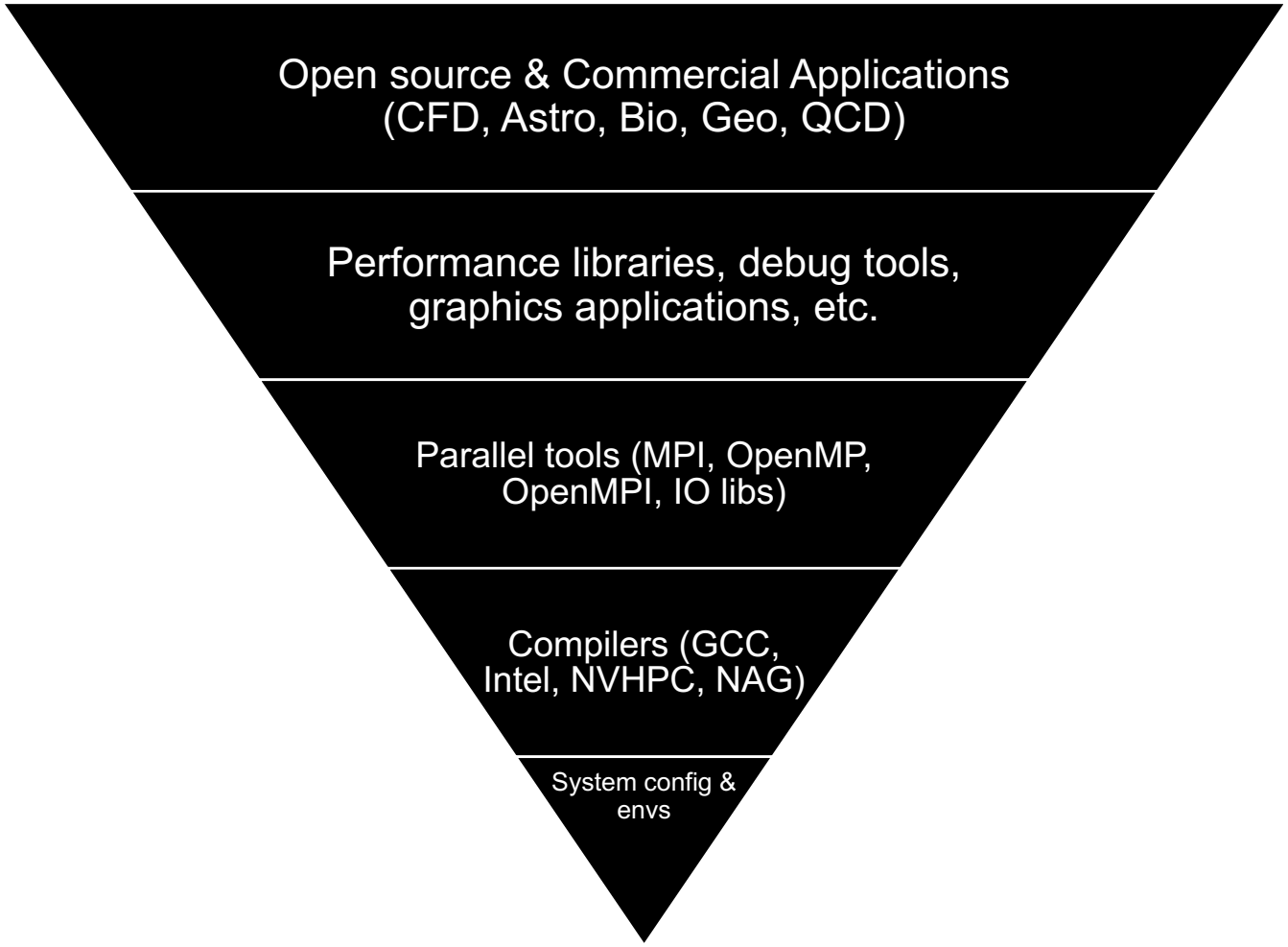
Software ecosystem on compute resources at LRZ



Getting user application ready for HPC

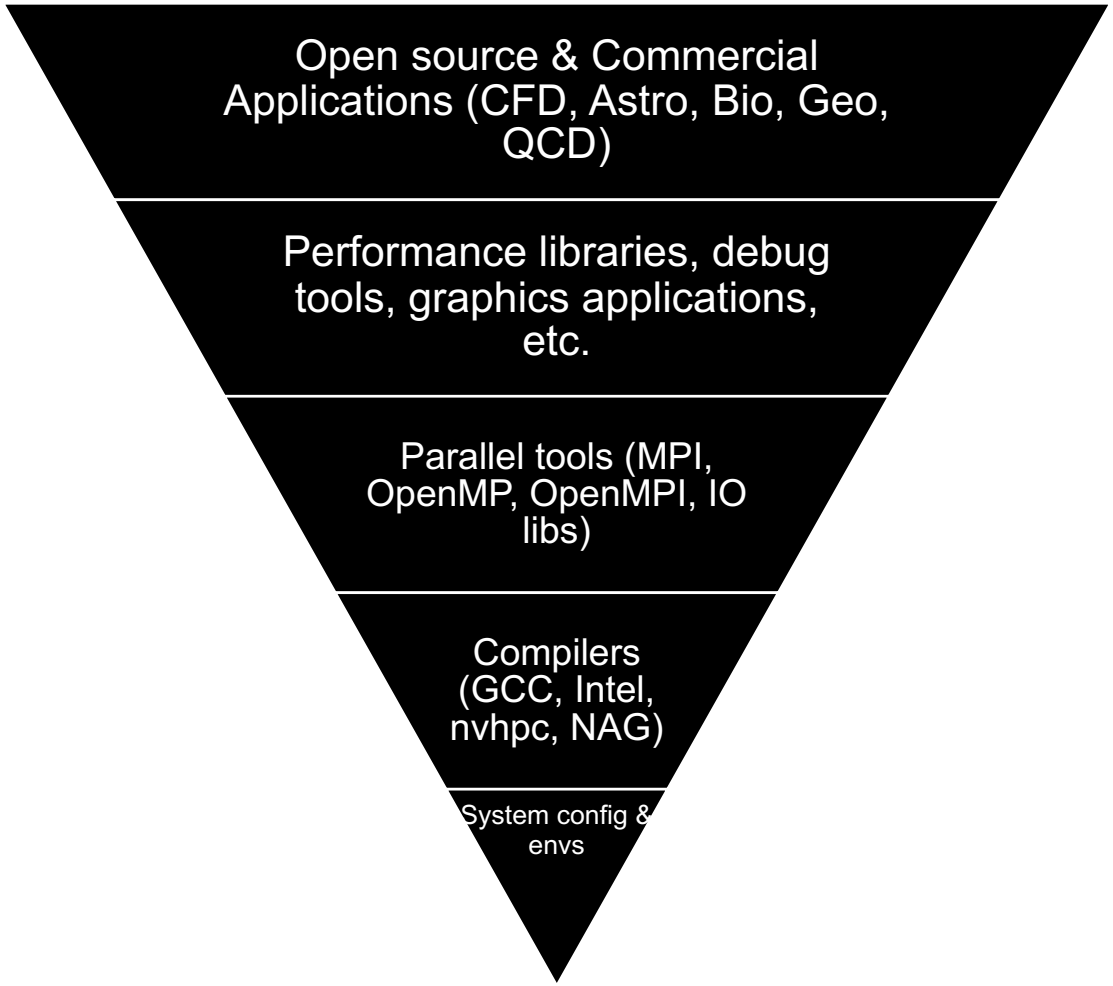


Software ecosystem/stack on compute resources



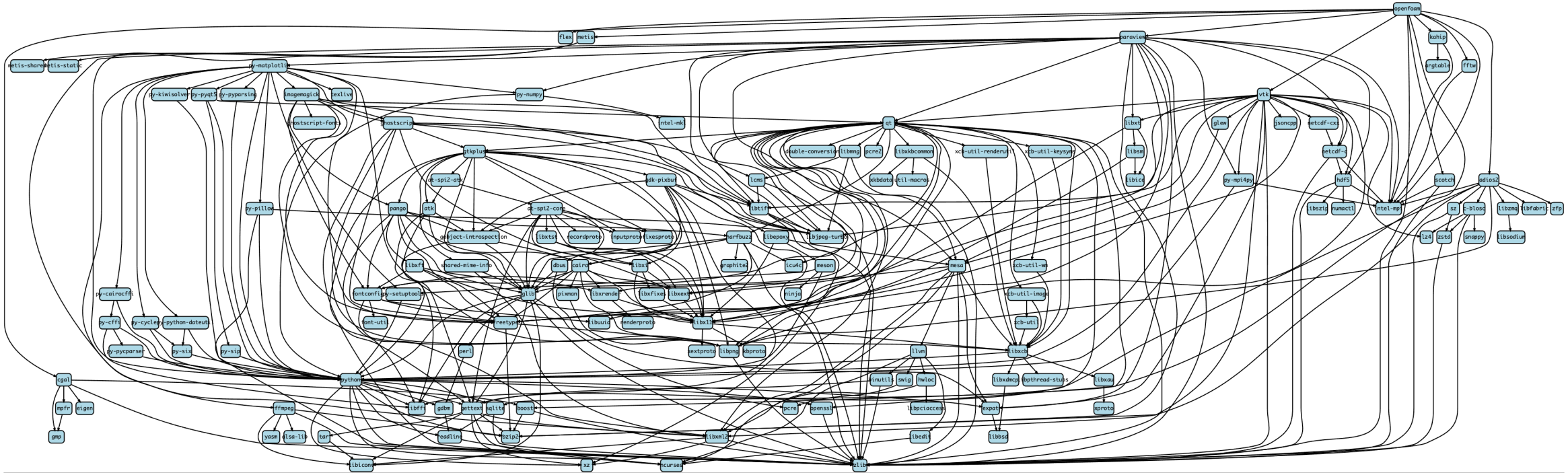
Software stack on compute resources

- In total about ~400 modules on each of these machines,
  - SuperMUC-NG
  - CoolMUC-2
  - CoolMUC-3
- Libraries and applications are build for,
  - Specific architectures
    - Skylake
    - Haswell
    - KNL
  - General build
    - x86_64
- A trimmed software stack on,
  - Housing systems
  - BEAST systems
  - RVS system
- Baring most commercial applications, **about >90%** of all software/libraries are provided with the help of **Spack**.



The dependency-hell

- A high-level application may just be the „tip of an iceberg“ when considering a feature-rich configuration of the software with all it‘s dependencies.
- Feature-rich CFD-Package **OpenFOAM** incl. vtk & paraview (with 140 dependencies) looks like,



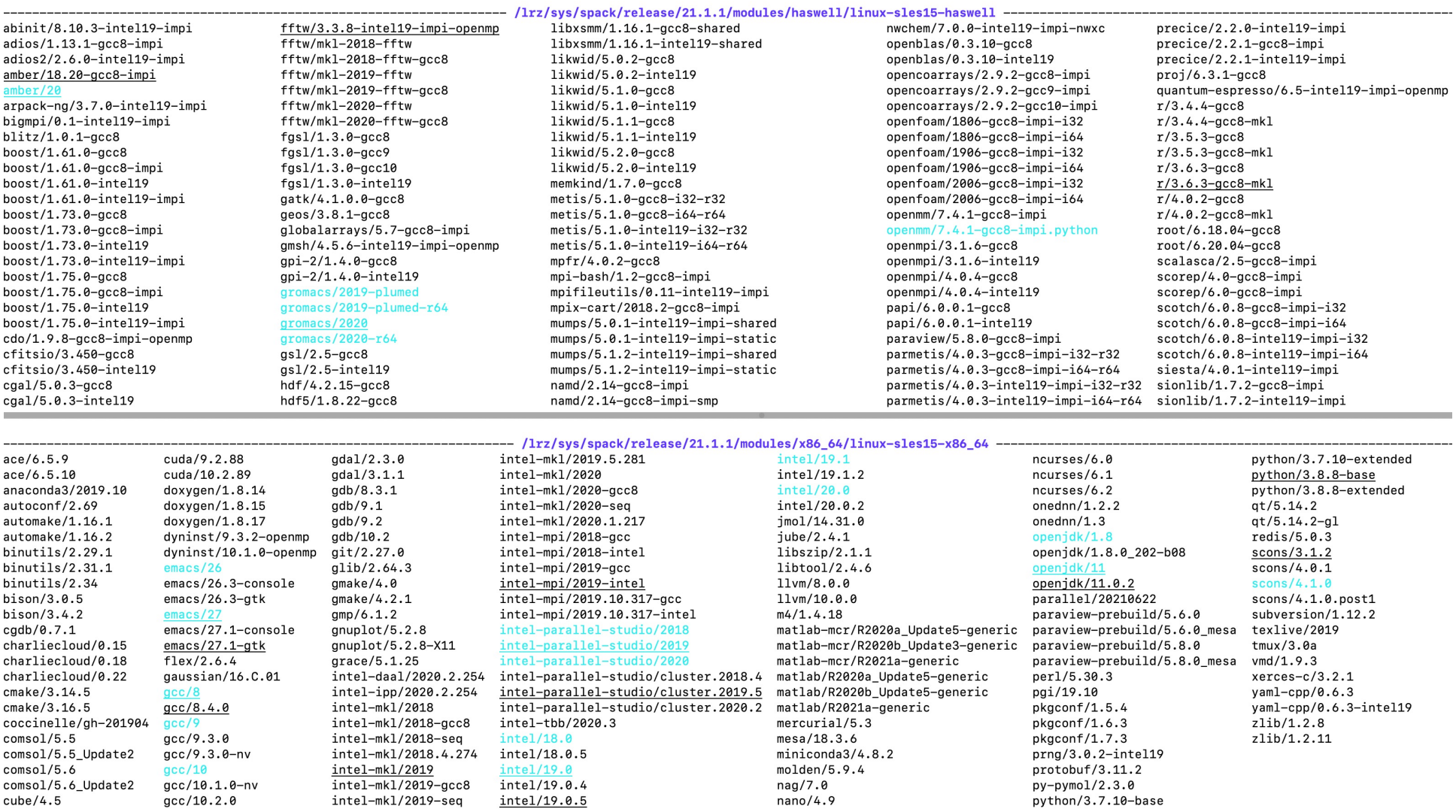
What is an Environmental Module?

- Modules provide a flexible way to configure and access various applications, compilers, tools and libraries dynamically by managing the shell environment

- In other words modules allow for the dynamic modification of environment variables such as;
  - library paths
  - binary paths
  - license server settings
  - application specific configurations

```
local PC :~> ssh lxlogin1.lrz.de

cm2login1:~> module list

1) admin/1.0   2) tempdir/1.0   3) lrz/1.0   4) spack/22.2.1
5) intel-oneapi-compilers/2021.4.0   6) intel-mkl/2020   7) intel-mpi/2019-intel

cm2login1:~> module show intel-oneapi-compilers

conflict        intel-oneapi-compilers
conflict        intel pgi nag intel-parallel-studio
prepend-path    LD_LIBRARY_PATH          /lrz/sys/spack/release/22.2.1/opt/…
prepend-path    PATH                     /lrz/sys/spack/release/22.2.1/opt/…
setenv          CC                       icx
setenv          CXX                      icpx
setenv          CPP                      {icx -E}
setenv          FC                       ifx
setenv          FORT_BUFFERED            true
setenv          FORT_BLOCKSIZE           c
module-whatis   compilers:fortran/c/c++/dpc++:icx:icpx:ifx:ifort:icc

cm2login1:~> module avail gcc

gcc/8.5.0  gcc/9.4.0  gcc/10.3.0 gcc/11.2.0


cm2login1:~> module load gcc
```

Screenshot of modules in the software stack



```
------------------------------------------------------------------------ /lrz/sys/spack/release/21.1.1/modules/haswell/linux-sles15-haswell ------------------------------------------------------------------------
abinit/8.10.3-intel19-impi            fftw/3.3.8-intel19-impi-openmp      libxsmm/1.16.1-gcc8-shared            nwchem/7.0.0-intel19-impi-nwxc      precice/2.2.0-intel19-impi
adios/1.13.1-gcc8-impi                fftw/mkl-2018-fftw                  libxsmm/1.16.1-intel19-shared         openblas/0.3.10-gcc8                precice/2.2.1-gcc8-impi
adios2/2.6.0-intel19-impi             fftw/mkl-2018-fftw-gcc8             likwid/5.0.2-gcc8                     openblas/0.3.10-intel19             precice/2.2.1-intel19-impi
amber/18.20-gcc8-impi                 fftw/mkl-2019-fftw                  likwid/5.0.2-intel19                  opencoarrays/2.9.2-gcc8-impi        proj/6.3.1-gcc8
amber/20                              fftw/mkl-2019-fftw-gcc8             likwid/5.1.0-gcc8                     opencoarrays/2.9.2-gcc9-impi        quantum-espresso/6.5-intel19-impi-openmp
arpack-ng/3.7.0-intel19-impi          fftw/mkl-2020-fftw                  likwid/5.1.0-intel19                  opencoarrays/2.9.2-gcc10-impi       r/3.4.4-gcc8
bigmpi/0.1-intel19-impi               fftw/mkl-2020-fftw-gcc8             likwid/5.1.1-gcc8                     openfoam/1806-gcc8-impi-i32         r/3.4.4-gcc8-mkl
blitz/1.0.1-gcc8                      fgsl/1.3.0-gcc8                     likwid/5.1.1-intel19                  openfoam/1806-gcc8-impi-i64         r/3.5.3-gcc8
boost/1.61.0-gcc8                     fgsl/1.3.0-gcc9                     likwid/5.2.0-gcc8                     openfoam/1906-gcc8-impi-i32         r/3.5.3-gcc8-mkl
boost/1.61.0-gcc8-impi                fgsl/1.3.0-gcc10                    likwid/5.2.0-intel19                  openfoam/1906-gcc8-impi-i64         r/3.6.3-gcc8
boost/1.61.0-intel19                  fgsl/1.3.0-intel19                  memkind/1.7.0-gcc8                    openfoam/2006-gcc8-impi-i32         r/3.6.3-gcc8-mkl
boost/1.61.0-intel19-impi             gatk/4.1.0.0-gcc8                   metis/5.1.0-gcc8-i32-r32              openfoam/2006-gcc8-impi-i64         r/4.0.2-gcc8
boost/1.73.0-gcc8                     geos/3.8.1-gcc8                     metis/5.1.0-gcc8-i64-r64              openmm/7.4.1-gcc8-impi              r/4.0.2-gcc8-mkl
boost/1.73.0-gcc8-impi                globalarrays/5.7-gcc8-impi          metis/5.1.0-intel19-i32-r32           openmm/7.4.1-gcc8-impi.python       root/6.18.04-gcc8
boost/1.73.0-intel19                  gmsh/4.5.6-intel19-impi-openmp      metis/5.1.0-intel19-i64-r64           openmpi/3.1.6-gcc8                  root/6.20.04-gcc8
boost/1.73.0-intel19-impi             gpi-2/1.4.0-gcc8                    mpfr/4.0.2-gcc8                       openmpi/3.1.6-intel19               scalasca/2.5-gcc8-impi
boost/1.75.0-gcc8                     gpi-2/1.4.0-intel19                 mpi-bash/1.2-gcc8-impi                openmpi/4.0.4-gcc8                  scorep/4.0-gcc8-impi
boost/1.75.0-gcc8-impi                gromacs/2019-plumed                 mpifileutils/0.11-intel19-impi        openmpi/4.0.4-intel19               scorep/6.0-gcc8-impi
boost/1.75.0-intel19                  gromacs/2019-plumed-r64             mpix-cart/2018.2-gcc8-impi            papi/6.0.0.1-gcc8                   scotch/6.0.8-gcc8-impi-i32
boost/1.75.0-intel19-impi             gromacs/2020                        mumps/5.0.1-intel19-impi-shared       papi/6.0.0.1-intel19                scotch/6.0.8-gcc8-impi-i64
cdo/1.9.8-gcc8-impi-openmp            gromacs/2020-r64                    mumps/5.0.1-intel19-impi-static       paraview/5.8.0-gcc8-impi            scotch/6.0.8-intel19-impi-i32
cfitsio/3.450-gcc8                    gsl/2.5-gcc8                        mumps/5.1.2-intel19-impi-shared       parmetis/4.0.3-gcc8-impi-i32-r32    scotch/6.0.8-intel19-impi-i64
cfitsio/3.450-intel19                 gsl/2.5-intel19                     mumps/5.1.2-intel19-impi-static       parmetis/4.0.3-gcc8-impi-i64-r64    siesta/4.0.1-intel19-impi
cgal/5.0.3-gcc8                       hdf/4.2.15-gcc8                     namd/2.14-gcc8-impi                   parmetis/4.0.3-intel19-impi-i32-r32 sionlib/1.7.2-gcc8-impi
cgal/5.0.3-intel19                    hdf5/1.8.22-gcc8                    namd/2.14-gcc8-impi-smp               parmetis/4.0.3-intel19-impi-i64-r64 sionlib/1.7.2-intel19-impi

------------------------------------------------------------------------ /lrz/sys/spack/release/21.1.1/modules/x86_64/linux-sles15-x86_64 ------------------------------------------------------------------------
ace/6.5.9                 cuda/9.2.88               gdal/2.3.0                intel-mkl/2019.5.281                  intel/19.1                          ncurses/6.0                   python/3.7.10-extended
ace/6.5.10                cuda/10.2.89              gdal/3.1.1                intel-mkl/2020                        intel/19.1.2                        ncurses/6.1                   python/3.8.8-base
anaconda3/2019.10         doxygen/1.8.14            gdb/8.3.1                 intel-mkl/2020-gcc8                   intel/20.0                          ncurses/6.2                   python/3.8.8-extended
autoconf/2.69             doxygen/1.8.15            gdb/9.1                   intel-mkl/2020-seq                    intel/20.0.2                        onednn/1.2.2                  qt/5.14.2
automake/1.16.1           doxygen/1.8.17            gdb/9.2                   intel-mkl/2020.1.217                  jmol/14.31.0                        onednn/1.3                    qt/5.14.2-gl
automake/1.16.2           dyninst/9.3.2-openmp      gdb/10.2                  intel-mpi/2018-gcc                    jube/2.4.1                          openjdk/1.8                   redis/5.0.3
binutils/2.29.1           dyninst/10.1.0-openmp     git/2.27.0                intel-mpi/2018-intel                  libszip/2.1.1                       openjdk/1.8.0_202-b08         scons/3.1.2
binutils/2.31.1           emacs/26                  glib/2.64.3               intel-mpi/2019-gcc                    libtool/2.4.6                       openjdk/11                    scons/4.0.1
binutils/2.34             emacs/26.3-console        gmake/4.0                 intel-mpi/2019-intel                  llvm/8.0.0                          openjdk/11.0.2                scons/4.1.0
bison/3.0.5               emacs/26.3-gtk            gmake/4.2.1               intel-mpi/2019.10.317-gcc             llvm/10.0.0                         parallel/20210622             scons/4.1.0.post1
bison/3.4.2               emacs/27                  gmp/6.1.2                 intel-mpi/2019.10.317-intel           m4/1.4.18                           paraview-prebuild/5.6.0       subversion/1.12.2
cgdb/0.7.1                emacs/27.1-console        gnuplot/5.2.8             intel-parallel-studio/2018            matlab-mcr/R2020a_Update5-generic   paraview-prebuild/5.6.0_mesa  texlive/2019
charliecloud/0.15         emacs/27.1-gtk            gnuplot/5.2.8-X11         intel-parallel-studio/2019            matlab-mcr/R2020b_Update3-generic   paraview-prebuild/5.8.0       tmux/3.0a
charliecloud/0.18         flex/2.6.4                grace/5.1.25              intel-parallel-studio/2020            matlab-mcr/R2021a-generic           paraview-prebuild/5.8.0_mesa  vmd/1.9.3
charliecloud/0.22         gaussian/16.C.01          intel-daal/2020.2.254     intel-parallel-studio/cluster.2018.4  matlab/R2020a_Update5-generic       perl/5.30.3                   xerces-c/3.2.1
cmake/3.14.5              gcc/8                     intel-ipp/2020.2.254      intel-parallel-studio/cluster.2019.5  matlab/R2020b_Update5-generic       pgi/19.10                     yaml-cpp/0.6.3
cmake/3.16.5              gcc/8.4.0                 intel-mkl/2018            intel-parallel-studio/cluster.2020.2  matlab/R2021a-generic               pkgconf/1.5.4                 yaml-cpp/0.6.3-intel19
coccinelle/gh-201904      gcc/9                     intel-mkl/2018-gcc8       intel-tbb/2020.3                      mercurial/5.3                       pkgconf/1.6.3                 zlib/1.2.8
comsol/5.5                gcc/9.3.0                 intel-mkl/2018-seq        intel/18.0                            mesa/18.3.6                         pkgconf/1.7.3                 zlib/1.2.11
comsol/5.5_Update2        gcc/9.3.0-nv              intel-mkl/2018.4.274      intel/18.0.5                          miniconda3/4.8.2                    prng/3.0.2-intel19
comsol/5.6                gcc/10                    intel-mkl/2019            intel/19.0                            molden/5.9.4                        protobuf/3.11.2
comsol/5.6_Update2        gcc/10.1.0-nv             intel-mkl/2019-gcc8       intel/19.0.4                          nag/7.0                             py-pymol/2.3.0
cube/4.5                  gcc/10.2.0                intel-mkl/2019-seq        intel/19.0.5                          nano/4.9                            python/3.7.10-base
```

Providing software with Spack at LRZ

- In the past at LRZ …
- Software stack on LRZ HPC-systems used to be provided via the module system in a non-orchestrated way with hand-written TCL-files to make installations available: applications/libraries/tools/compilers
- Limitations:
  - Non-transparent or oblique conflicts and/or dependencies of packages
  - Non-transparent package-configs and build-variants
  - Builds often not reproducible (documentation issue)

- Since 2018 at LRZ …
- Most software (barring a few commercial software) we provide are installed using Spack.
- Advantages:
  - Spack Builds are self-documenting
    - Package-builds are typically reproducible
  - Spack-compiler wrappers inject compiler-flags for the target-architecture -> optimized software stack
  - Installation of many package-variants do not disturb each other -> many packages may peacefully coexist
  - Installation (fetch/configure/build/install/module-create) of the software is automized

Spack is one of many package-managers

- Functional Cross-Platform Package Managers:
  e.g Nix (NixOs), Gnu Guix (Gnu Guix Linux) … use hashes in install-dirs

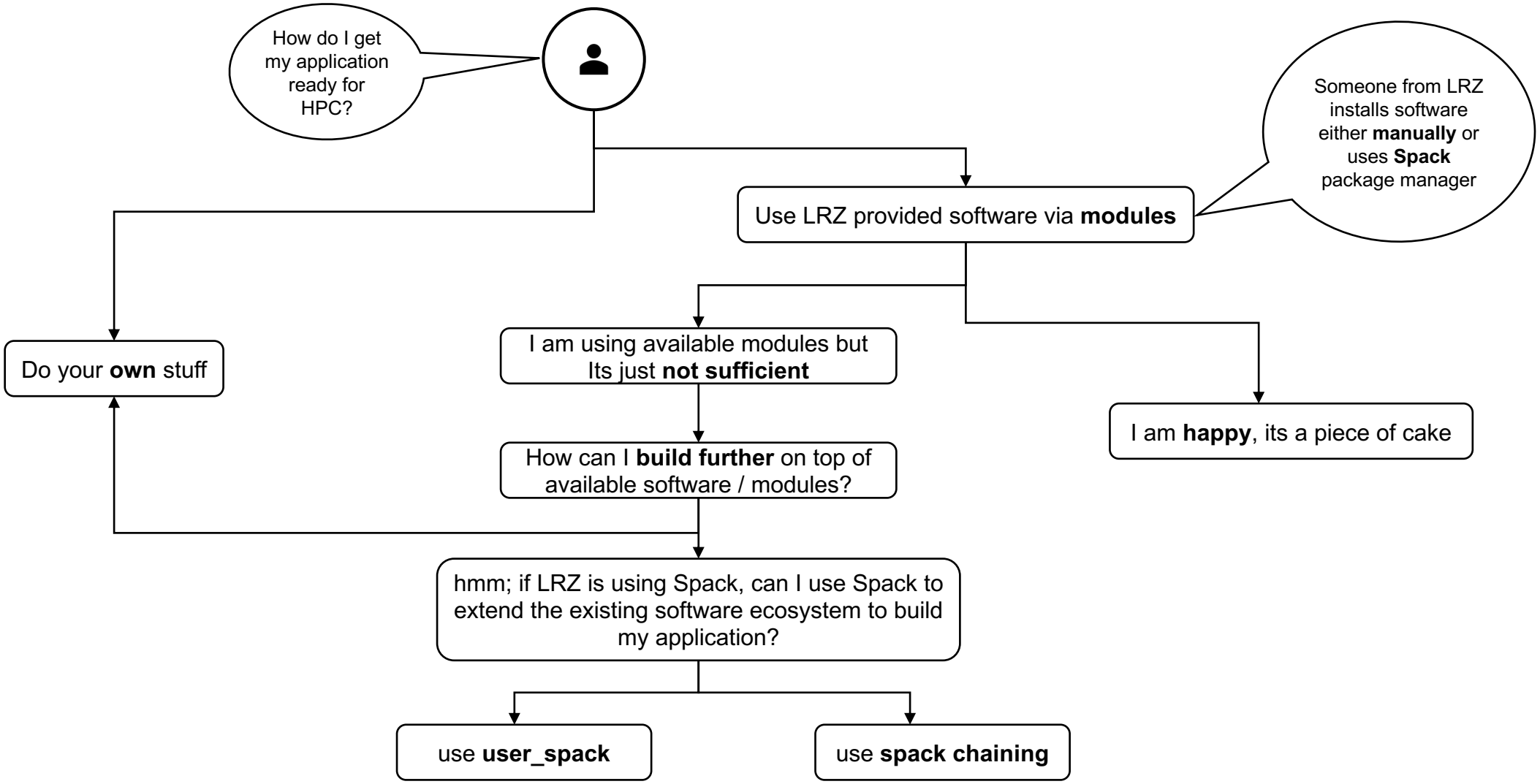
- Build-from-source Package Managers
  e.g. HomeBrew/LinuxBrew

- Package Managers for specific scripting languages
  e.g. Pip (Python), NPM (Javascript)

- Easy Build:
  installation framework for managing scientific software on HPC-systems

- Conda:
  popular binary package managers for Python and R (but also for other rpm–like packaging in user-space). Easy to use.
  In general no architecture optimized binaries, not targeted at HPC

Flow chart to get my application ready for HPC



If you want to use Spack,
you might want to know about it

What is Spack? In a nutshell

- Spack automates the fetch, configure and installation of scientific software.
- It‘s a package manager and not build tool, like cmake / Autotools.
- Spack works out of the box. Only few prerequisites from system side, like basic python, git, make, c/c++ compiler, etc.

- Simply clone Spack to get going.

```
>git clone https://github.com/spack/spack
Cloning into 'spack'...
remote: Enumerating objects: 355317, done.
remote: Total 355317 (delta 0), reused 0 (delta 0), pack-reused 355317
Receiving objects: 100% (355317/355317), 159.99 MiB | 23.13 MiB/s, done.
Resolving deltas: 100% (151370/151370), done.
Updating files: 100% (9087/9087), done.
>
>. spack/share/spack/setup-env.sh
>
```

```
>spack install zlib
==> Installing zlib-1.2.11-fy5vijvec3tmqbdnsh4q3wkmqsqpzb5a
==> No binary for zlib-1.2.11-fy5vijvec3tmqbdnsh4q3wkmqsqpzb5a found: installing from source
==> Fetching https://mirror.spack.io/_source-cache/archive/c3/c3e5e9fdd5004dcb542feda5ee4f0ff0744628baf8ed2dd5d66f8ca1197cb1a1.tar.gz
==> No patches needed for zlib
==> zlib: Executing phase: 'install'
==> zlib: Successfully installed zlib-1.2.11-fy5vijvec3tmqbdnsh4q3wkmqsqpzb5a
  Fetch: 0.29s.  Build: 2.18s.  Total: 2.48s.
[+] /dss/dsshome1/09/di36pex/demo_spack/spack/opt/spack/linux-sles15-haswell/gcc-8.4.0/zlib-1.2.11-fy5vijvec3tmqbdnsh4q3wkmqsqpzb5a
```

What is Spack? In a nutshell

- Spack can install many different variants of the same package, to name a few:
  - package-versions
  - different compilers
  - different MPI-implementations
  - different build options
- Most important terminology using Spack is "**spec**"
  - Spec is what comes after "spack install" command.
  - Specs refer to a particular build configuration of a package.
  - "Specs" are more than package name. It can contain the compiler, compiler version, architecture, compile options, and dependency options for a build.
- Installation locations are separated via unique hashes
  - installations may peacefully coexist (dynamic linking with RPATH)
- The installation location of any package will also contain,
  - dump (in form of text files) of environment during installation, output from installation, configure arguments, and concretized spack-specs

- Install a package
  - ➢ spack install hdf5
- Install a particular version by appending @
  - ➢ spack install hdf5@1.12.1
- Specify a compiler (and its version), with %
  - ➢ spack install hdf5%gcc@11.2.0
- Add special boolean compile-time options with +
  - ➢ spack install hdf5@1.12.1%gcc@11.2.0 +fortran +hl
- Add compiler flags using the conventional names
  - ➢ spack install hdf5%gcc@11.2.0 +cxx cppflags="-O3 -floop-block"
- Add micro-architecture with target (for cross compiling)
  - ➢ $ spack install hdf5@1.12.1%gcc target=skylake_avx512

Finding version and variants of a package

- Spack will fetch all the information from a “package file” of a package.
  - ➢ Spack info <package name>

```
>spack info hdf5
CMakePackage:   hdf5

Description:
    HDF5 is a data model, library, and file format for storing and managing
    data. It supports an unlimited variety of datatypes, and is designed for
    flexible and efficient I/O and for high volume and complex data.

Homepage: https://portal.hdfgroup.org

Maintainers: @lrknox @brtnfld @byrnHDF @ChristopherHogan @epourmal @gheber @hyoklee @lkurz @soumagne

Externally Detectable:
    False

Tags:
    e4s

Preferred version:
    1.12.1          https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.12/hdf5-1.12.1/src/hdf5-1.12.1.tar.gz

Safe versions:
    develop-1.13    [git] https://github.com/HDFGroup/hdf5.git on branch develop
    1.8.10          https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8/hdf5-1.8.10/src/hdf5-1.8.10.tar.gz

Deprecated versions:
    None

Variants:
    Name [Default]              When    Allowed values          Description
    ==========================  ====    ====================    ==========================================

    api [default]               --      default, v114, v112,    Choose api compatibility for earlier version
                                        v110, v18, v16
    build_type [RelWithDebInfo] --      Debug, Release,         CMake build type
                                        RelWithDebInfo,
                                        MinSizeRel
    cxx [off]                   --      on, off                 Enable C++ support
    fortran [off]               --      on, off                 Enable Fortran support
    hl [off]                    --      on, off                 Enable the high-level library
    ipo [off]                   --      on, off                 CMake interprocedural optimization
    java [off]                  --      on, off                 Enable Java support
    mpi [on]                    --      on, off                 Enable MPI support
    threadsafe [off]            --      on, off                 Enable thread-safe capabilities
    tools [on]                  --      on, off                 Enable building tools

Installation Phases:
    cmake    build    install

Build Dependencies:
    cmake  java  mpi  numactl  szip  zlib

Link Dependencies:
    mpi  numactl  szip  zlib

Run Dependencies:
    java  pkgconfig
```

Concretization: dependency tree of a package

- Spack will fetch all the information from a “package file” of a package.
  - ➢ Spack info <package name>
- Listing a dependency graph before you go ahead with an installation
  - ➢ Spack spec –INtL <package name>

➢ Spack spec gives you full concretized map of the package

```
>spack spec -lINt hdf5 +fortran +hl
[Input spec
--------------------------------
 -   [    ]  .hdf5+fortran+hl

Concretized
--------------------------------
 -   qhat2bv  [    ]  builtin.hdf5@1.12.1%gcc@8.4.0~cxx+fortran+hl~ipo~java+mpi+shared~szip~threadsa
 -   i5dqbwv  [b   ]      ^builtin.cmake@3.22.2%gcc@8.4.0~doc+ncurses+openssl+ownlibs~qt build_type=
 -   lg6uazi  [bl  ]          ^builtin.ncurses@6.2%gcc@8.4.0~symlinks+termlib abi=none arch=linux-sl
 -   daqug65  [b r ]              ^builtin.pkgconf@1.8.0%gcc@8.4.0 arch=linux-sles15-haswell
 -   2zfrt75  [bl  ]          ^builtin.openssl@1.1.1m%gcc@8.4.0~docs certs=system arch=linux-sles15-
 -   zveogua  [b r ]              ^builtin.perl@5.34.0%gcc@8.4.0+cpanm+shared+threads arch=linux-sle
 -   t5a6kgl  [bl  ]                  ^builtin.berkeley-db@18.1.40%gcc@8.4.0+cxx~docs+stl patches=b2
 -   dpfohvz  [bl  ]                  ^builtin.bzip2@1.0.8%gcc@8.4.0~debug~pic+shared arch=linux-sle
 -   ji7jj7r  [b   ]                      ^builtin.diffutils@3.8%gcc@8.4.0 arch=linux-sles15-haswell
 -   rtusfmn  [bl  ]                          ^builtin.libiconv@1.16%gcc@8.4.0 libs=shared,static ar
 -   b3okknr  [bl  ]                  ^builtin.gdbm@1.19%gcc@8.4.0 arch=linux-sles15-haswell
 -   tjsehf3  [bl  ]                      ^builtin.readline@8.1%gcc@8.4.0 arch=linux-sles15-haswell
[+]  fy5vijv  [bl  ]                  ^builtin.zlib@1.2.11%gcc@8.4.0+optimize+pic+shared arch=linux-
 -   sklqwij  [bl  ]      ^builtin.numactl@2.0.14%gcc@8.4.0 patches=4e1d78cbbb85de625bad28705e748856
f599cfabf0740518b91ec8daaf18e8f288b19adaae5364dc1f6b2296 arch=linux-sles15-haswell
 -   kyq3asj  [b   ]          ^builtin.autoconf@2.69%gcc@8.4.0 patches=35c449281546376449766f92d49fc
5bac3b62daa0ff688ab4d508d71dbd2f4f8d7e2a02321926346161bf3ee arch=linux-sles15-haswell
 -   lx4bpfg  [b r ]              ^builtin.m4@1.4.19%gcc@8.4.0+sigsegv patches=9dc5fbd0d5cb1037ab1e6
89 arch=linux-sles15-haswell
 -   6zmn6dv  [bl  ]                  ^builtin.libsigsegv@2.13%gcc@8.4.0 arch=linux-sles15-haswell
 -   y77iduv  [b   ]          ^builtin.automake@1.16.5%gcc@8.4.0 arch=linux-sles15-haswell
 -   kyyaw3y  [b   ]          ^builtin.libtool@2.4.6%gcc@8.4.0 arch=linux-sles15-haswell
 -   l7ukpnd  [bl  ]      ^builtin.openmpi@4.1.2%gcc@8.4.0~atomics~cuda~cxx~cxx_exceptions+gpfs~inte
e+vt+wrapper-rpath fabrics=none schedulers=none arch=linux-sles15-haswell
 -   gzlecom  [bl  ]          ^builtin.hwloc@2.7.0%gcc@8.4.0~cairo~cuda~gl~libudev+libxml2~netloc~nv
 -   qxup3jg  [bl  ]              ^builtin.libpciaccess@0.16%gcc@8.4.0 arch=linux-sles15-haswell
 -   kg3gaar  [b   ]                  ^builtin.util-macros@1.19.3%gcc@8.4.0 arch=linux-sles15-haswel
 -   wwlegep  [bl  ]              ^builtin.libxml2@2.9.12%gcc@8.4.0~python arch=linux-sles15-haswell
 -   4syc2vd  [bl  ]                  ^builtin.xz@5.2.5%gcc@8.4.0~pic libs=shared,static arch=linux-
 -   4sogqx7  [bl  ]          ^builtin.libevent@2.1.12%gcc@8.4.0+openssl arch=linux-sles15-haswell
 -   jitrjio  [  r ]          ^builtin.openssh@8.8p1%gcc@8.4.0 arch=linux-sles15-haswell
 -   jkppbin  [bl  ]              ^builtin.libedit@3.1-20210216%gcc@8.4.0 arch=linux-sles15-haswell
```

Useful Spack commands

```
$ spack help
usage: spack [-hkV] [--color {always,never,auto}] COMMAND ...

A flexible package manager that supports multiple versions,
configurations, platforms, and compilers.

These are common spack commands:

query packages:
  list                  list and search available packages
  info                  get detailed information on a particular package
  find                  list and search installed packages

build packages:
  install               build and install packages
  uninstall             remove installed packages
  gc                    remove specs that are now no longer needed
  spec                  show what would be installed, given a spec

configuration:
  external              manage external packages in Spack configuration

environments:
  env                   manage virtual environments
  view                  project packages to a compact naming scheme on the filesystem.

create packages:
  create                create a new package file
  edit                  open package files in $EDITOR

system:
  arch                  print architecture information about this machine
  audit                 audit configuration files, packages, etc.
  compilers             list available compilers
```

• Spack in User space – “user_spack” module

- Chaining existing Installation into your own Spack Environment
- How do I activate user_spack?
  - module load user_spack
- Why do I use it?
  - making use of already installed packages via chaining of upstream-location (lrzs/sys/spack/x/y)
  - avoids recompiling low level packages in many situations
  - has working defaults configurated for some essential dependencies (e.g. MPI)
- What does loading “user_spack” module do actually?
  - loads a setup script that adds the spack command to your shell.
  - spack version matches the version the default software stack has been built with
  - LRZ specific configurations are preconfigured for you

```
>spack spec -INlt hdf5 +fortran ~threadsafe
Input spec
--------------------------------
 -   [    ]  .hdf5+fortran~threadsafe

Concretized
--------------------------------
 -   ohhl2i4  [    ]  builtin.hdf5@1.10.7%gcc@11.2.0~cx
h=linux-sles15-haswell
[^]  kdeymh2  [b   ]      ^builtin.cmake@3.21.4%gcc@11.
[^]  47oirry  [bl  ]          ^builtin.ncurses@6.2%gcc@
[^]  6dqwgxm  [b r ]              ^builtin.pkgconf@1.8.
[^]  oeopnpj  [bl  ]          ^builtin.openssl@1.1.1l%g
[^]  rken265  [b r ]              ^builtin.perl@5.34.0%
[^]  dp5zt2x  [bl  ]                  ^builtin.berkeley-
05aff41de522 arch=linux-sles15-haswell
[^]  ybteog6  [bl  ]                  ^builtin.bzip2@1.
[^]  gweivqh  [b   ]                      ^builtin.diff
[^]  q6m4kz6  [bl  ]                          ^builtin.
```

• Installing software with user_spack

- HDF5 dependency tree is shown in the image.
- The library itself is not yet installed (-).
- But all of its dependencies are already available via the upstream LRZ installation ([^]).
- A package that you have installed locally in your home directory with Spack is marked by [+].

```
>spack spec -INlt hdf5 +fortran ~threadsafe
Input spec
--------------------------------
 -   [    ]  .hdf5+fortran~threadsafe

Concretized
--------------------------------
 -   ohhl2i4  [    ]  builtin.hdf5@1.10.7%gcc@11.2.0~cx
h=linux-sles15-haswell
[^]  kdeymh2  [b   ]      ^builtin.cmake@3.21.4%gcc@11.
[^]  47oirry  [bl  ]          ^builtin.ncurses@6.2%gcc@
[^]  6dqwgxm  [b r ]              ^builtin.pkgconf@1.8.
[^]  oeopnpj  [bl  ]          ^builtin.openssl@1.1.1l%g
[^]  rken265  [b r ]              ^builtin.perl@5.34.0%
[^]  dp5zt2x  [bl  ]                  ^builtin.berkeley-
05aff41de522 arch=linux-sles15-haswell
[^]  ybteog6  [bl  ]                  ^builtin.bzip2@1.
[^]  gweivqh  [b   ]                      ^builtin.diff
[^]  q6m4kz6  [bl  ]                          ^builtin.
```

• Installing your software with user_spack

- Dependencies are chained from the LRZ installation
  - see /dss/…/lrz/sys/spack/… paths
- Spack checks if the source tar ball is available in the LRZ cache
- If not present, tar files can be download from external site
- installation location of the library in $HOME/spack/opt/...

```
>spack install hdf5 +fortran ~threadsafe
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/pkg
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/be
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/li
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/zl
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/au
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/cm
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/au
[+] /dss/dsshome1/lrz/sys/spack/release/22.2.1/opt/haswell/nu
==> Installing hdf5-1.10.7-ohhl2i4g4ztco53okwghc5zcre332lk2
==> No binary for hdf5-1.10.7-ohhl2i4g4ztco53okwghc5zcre332lk
==> Fetching file:///lrz/sys/spack/cache/_source-cache/archive
==> Ran patch() for hdf5
==> hdf5: Executing phase: 'cmake'
==> hdf5: Executing phase: 'build'
==> hdf5: Executing phase: 'install'
==> hdf5: Successfully installed hdf5-1.10.7-ohhl2i4g4ztco53o
  Fetch: 0.04s.  Build: 2m 2.08s.  Total: 2m 2.12s.
[+] /dss/dsshome1/09/di36pex/spack/opt/linux-sles15-haswell/h
```

Generating Modules for your software

- We have configured Spack such that modules will only be generated for explicitly installed packages.
- Modules are installed in $HOME/spack/modules/$LRZ_INSTRSET/<architecture>/<package>/<version>
- The path to the modules is added to $MODULEPATH when you load the user_spack module, but only if it already exists. You might want to reload user_spack.
- If the does not exist,
  - > module use <path to local modules>
- To check if the path exist in the list of module paths by,
  - > module use

```
>spack module tcl refresh hdf5
==> You are about to regenerate tcl mod

-- linux-sles15-haswell / gcc@11.2.0 --
ohhl2i4 hdf5@1.10.7

==> Do you want to proceed? [y/n] y
==> Regenerating tcl module files
```

Configuring your Spack Instance

- LRZ provides a configuration that is very similar to the one the software stack was built with.
- You may want to change some or all of these settings to serve your needs, e.g. for the package selection, generation of modules, etc.
- Your individual configuration files are stored in the directory ~/.spack/.
  - > spack config edit repos
  - > spack config edit config
  - > spack config edit modules
- User config files take precedence over system provided config file, that is they are loaded after the system config files and overwrite their settings.

Example: Generating modules for upstream packages

- For packages that have been implicitly installed in the upstream spack (LRZ installed) stack no modules are generated by default.
- You can configure your `user_spack` such that you generate modules in your $HOME directory.
- You may want to change some or all of these settings to serve your needs, e.g. for the package selection, generation of modules, etc.
  - spack config edit modules
  - cat ~/.spack/modules.yaml

```
modules:
  tcl:
    blacklist_implicits: False
    hash_length: 7
```

- Now modules are generated for all installed modules and each newly created module gets a hash suffix of length 7 to avoid naming conflicts.
- Modules are generated with
  - spack module tcl refresh --upstream-modules

Configuring your Spack Instance

- **Adding compilers**
- Spack searches for compilers on your machine automatically the first time it is run. It does this by inspecting your $PATH
- One could use “spack compiler add” command,
  - spack compiler add /path/to/my_compiler
  - spack compiler list
- For example the output of spack compiler list could look like,

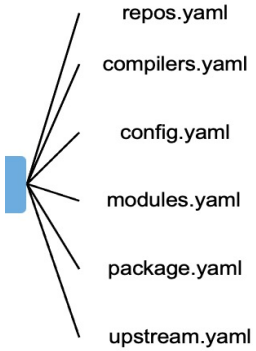
```
==> Available compilers

-- gcc sles15-x86_64 ---------------------------------------------
gcc@8.4.0  gcc@7.5.0


-- intel sles15-x86_64 -------------------------------------------
intel@19.0.5.281
```

- **Spack package repositories**
- Spack supports external package repositories
  - Separate directories of package files
- Many reasons for doing this,
  - You want to write a package file for an in-house code that you may not want to release publicly.
  - Overwrite default package files with site specific versions or restrictions
- One could use “spack repo create” command,
  - spack repo create /path/to/my_repo
  - spack repo add my_repo
  - spack repo list
- This will show 2 package repositories.
  - my_repo /path/to/my_repo builtin
  - spack/var/spack/repos/builtin

Chaining Spack Installations

- You can point your Spack installation to another installation to use any packages that are installed there.
- To register the other Spack instance, you can add it as an entry to upstreams.yaml

```
upstreams:
  spack-instance-1:
    install_tree: /path/to/other/spack/opt/spack
  spack-instance-2:
    install_tree: /path/to/another/spack/opt/spack
```




```
>spack spec -INlt hdf5 +fortran~hl
Input spec
--------------------------------
 -   [    ]  .hdf5+fortran~hl

Concretized
--------------------------------
 -   zt4lxrl  [    ]  fixes015x.hdf5@1.10.7%gcc@8.4.0~cxx~debug+fc
[^]  cyojcvv  [bl   ]      ^fixes015x.intel-mpi@2019.8.254%gcc@8.4.
[^]  wz47lgr  [bl   ]      ^builtin.numactl@2.0.12%gcc@8.4.0 arch=l
[^]  jns7liw  [b    ]          ^builtin.autoconf@2.69%gcc@8.4.0 arc
[^]  6vxvnrt  [b r  ]              ^builtin.m4@1.4.18%gcc@8.4.0+sig
c8 arch=linux-sles15-haswell
[^]  gv36h32  [bl   ]                  ^builtin.libsigsegv@2.12%gcc
[^]  bhpjih4  [b r  ]              ^builtin.perl@5.30.3%gcc@8.4.0+c
[^]  szzheyp  [bl   ]                  ^builtin.gdbm@1.18.1%gcc@8.4
[^]  3kfx6pu  [bl   ]                      ^builtin.readline@8.0%gc
[^]  6qhv5ta  [bl   ]                          ^fixes015x.ncurses@6
[^]  cfijkws  [b    ]                              ^builtin.pkgconf
[^]  zzoup2h  [b    ]          ^builtin.automake@1.16.2%gcc@8.4.0 a
[^]  4nya677  [b    ]          ^builtin.libtool@2.4.6%gcc@8.4.0 arc
[^]  m2bfsoy  [bl   ]      ^builtin.zlib@1.2.11%gcc@8.4.0+optimize+
```

Spack Environments

- A spack environment is used to group together a set of specs for the purpose of building, rebuilding and deploying in a coherent fashion.

- Lets say you want to create an spack environment for your group / team / project.

  - ➢ spack env create XYZ_env

  - Spack then creates the directory var/spack/environments/ XYZ_env
  - Within this directory, a file spack.yaml and the hidden directory .spack-env will exists
  - Spack stores metadata in the .spack-env directory. User interaction will occur through the spack.yaml
  - In short, **spack.yaml describes project requirements**
  - When the environment is concretized, Spack will create a file spack.lock. This file describes exactly what versions/configurations were installed, allows them to be reproduced.

  - You can give this file to any one in the project and he would get the exact same customized sets of packages installed, without any differences. A very robust reproducible software environment!

```
>cat recipes/x86_64/python-extended.yaml
spack:
  include:
  - ./config/
  specs:
  - python@3.8.11
  - py-mpi4py
  - py-scipy
  - py-numpy
  - py-pip
  - py-flake8
  - py-pylint
  - py-autopep8
  - py-ipython
  view:
    default:
      root: ../../views/python/3.8.11-extended
```

Questions?