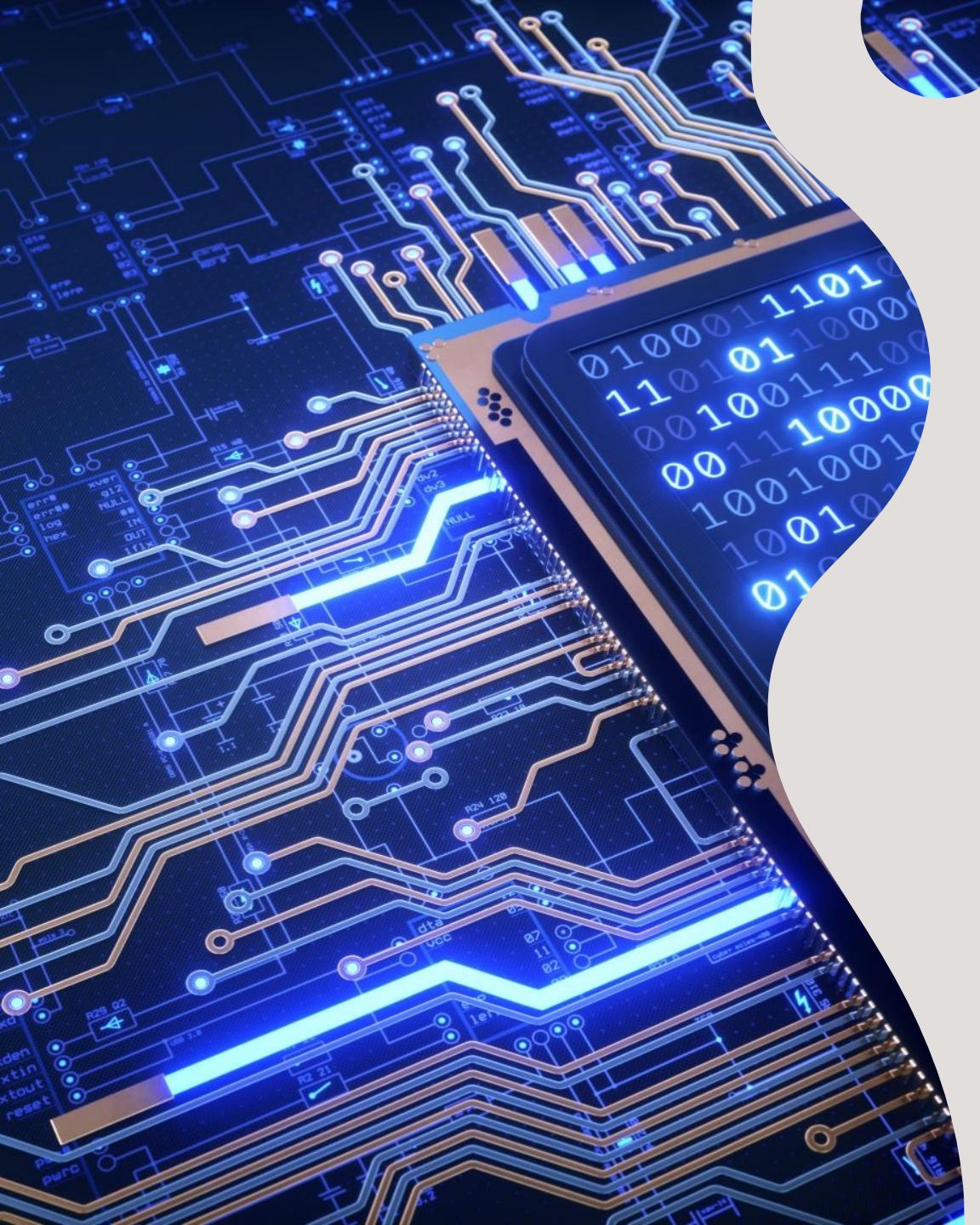


Introduction to spack

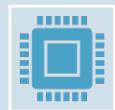
10/17/23 by Kevin Fotso



What is spack?



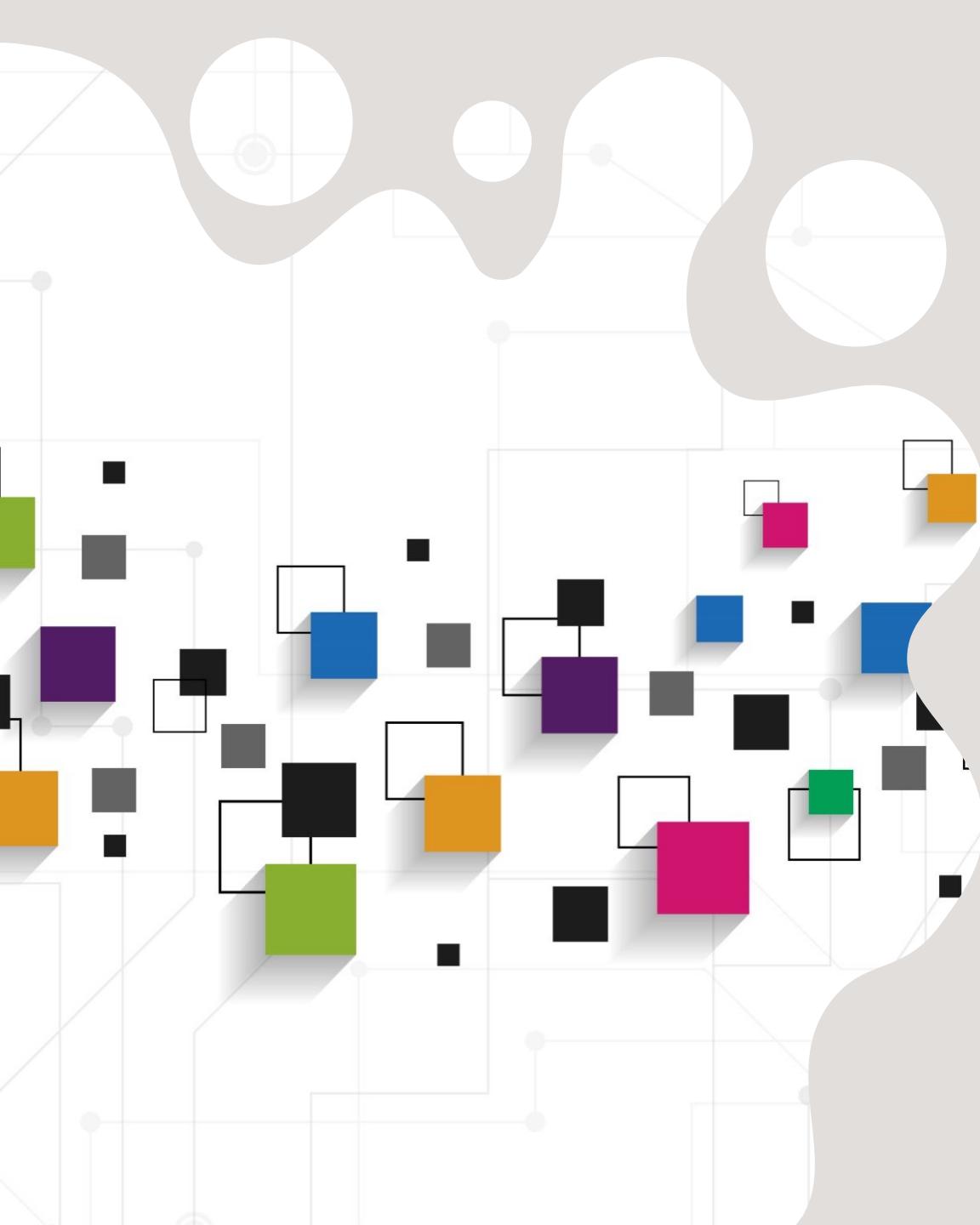
A package manager used in HPC.



Can be used on basic servers or even personal computers as well (Linux, MacOS)



Allows to build a package with multiple versions , configs, compilers, architecture



Context of spack creation

Complexity of HPC software is far greater than current software management tools.

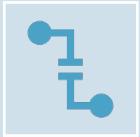
Some very complex software might require weeks to be installed properly (e.g. Trilinos)

Apps might require specific versions of compiler or dependency libraries

Benefits of spack (1)



Spack was developed at Lawrence Livermore National Laboratory.

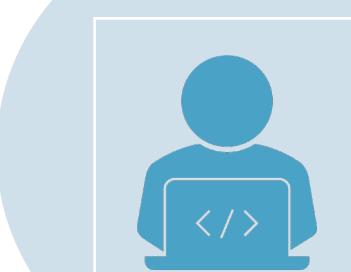


Builds can coexist on the same system and installed packages can find their dependencies, regardless of the environment.

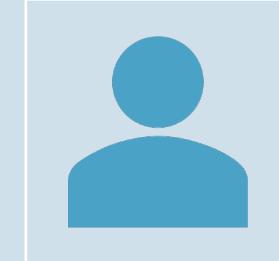


Installing the default version of a package is simple.

Benefits of spack (2)



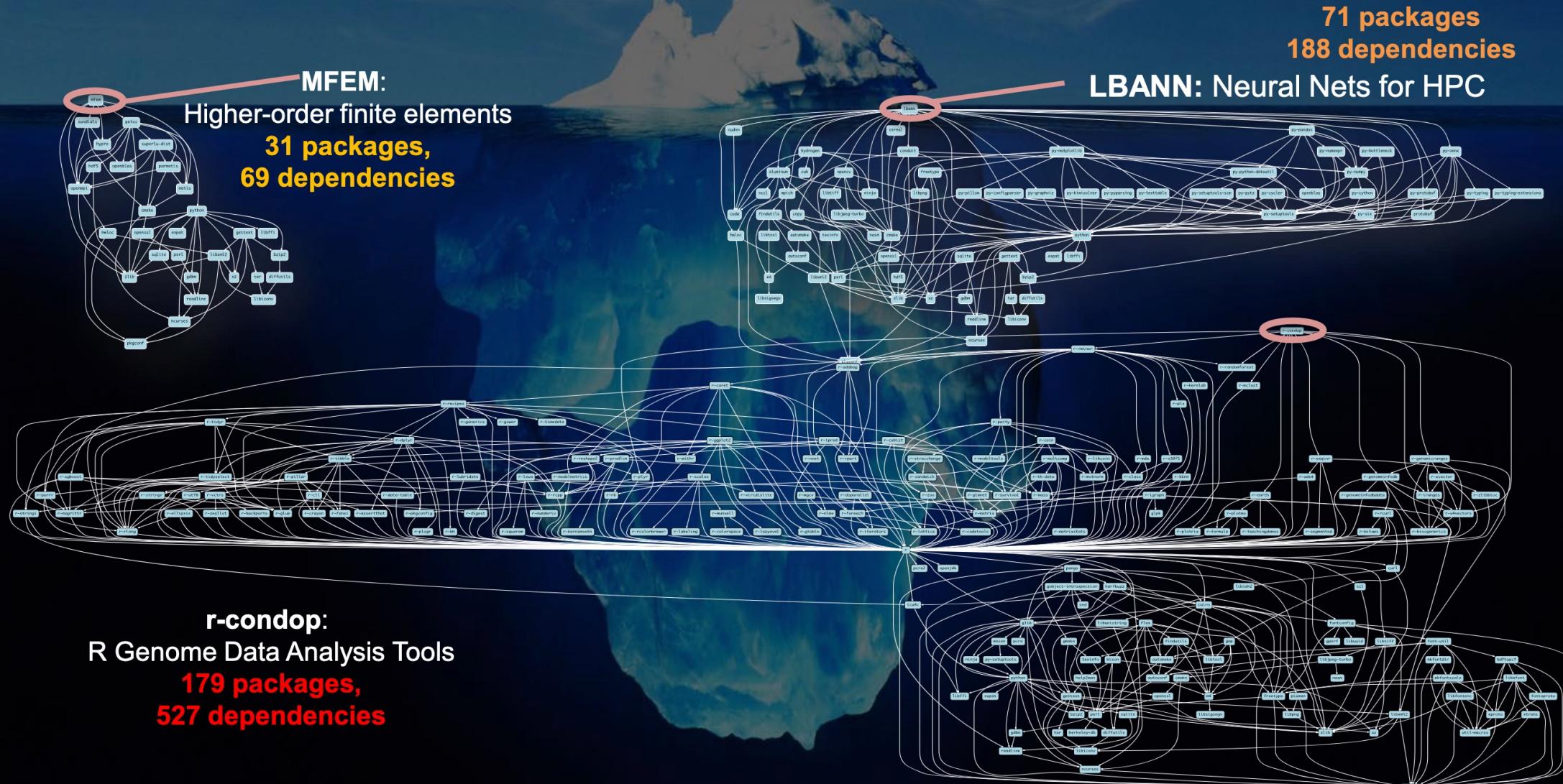
INSTALLATION CAN BE CUSTOMIZED
[VERSION, COMPILER, DEPENDENCIES]



CAN ALLOW USERS TO CREATE THEIR
OWN PACKAGES

Incentive for slack

Modern scientific codes rely on icebergs of dependency libraries



Source: <https://extremecomputingtraining.anl.gov/wp-content/uploads/sites/96/2023/08/ATPESC-2023-Track-3-Talk-4-Becker-Spack.pdf>

What is a spack package? (1)



They are installation scripts,
recipe for building the
software.



For example where to find
them (metadata), the
dependencies, the versions,
the variants etc ...

What is a spack package (2)

Spack packages are *parameterized* using the spec syntax
Python DSL defines many ways to build

```
from spack import *

class Kripke(CMakePackage):
    """Kripke is a simple, scalable, 3D Sn deterministic particle transport mini-app."""

    homepage = "https://computation.llnl.gov/projects/co-design/kripke"
    url     = "https://computation.llnl.gov/projects/co-design/download/kripke-openmp-1.1.tar.gz"

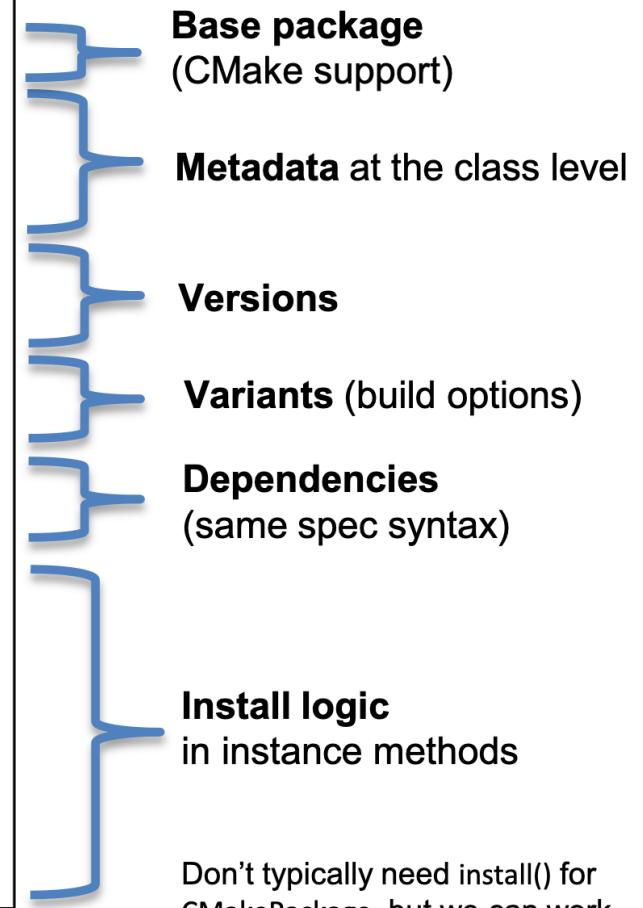
    version('1.2.3', sha256='3f7f2eef0d1ba5825780d626741eb0b3f026a096048d7ec4794d2a7dfbe2b8a6')
    version('1.2.2', sha256='eaf9ddf562416974157b34d00c3a1c880fc5296fce2aa2efa039a86e0976f3a3')
    version('1.1', sha256='232d74072fc7b848fa2adc8a1bc839ae8fb5f96d50224186601f55554a25f64a')

    variant('mpi', default=True, description='Build with MPI.')
    variant('openmp', default=True, description='Build with OpenMP enabled.')

    depends_on('mpi', when='+mpi')
    depends_on('cmake@3.0:', type='build')

    def cmake_args(self):
        return [
            '-DENABLE_OPENMP=%s' % ('+openmp' in self.spec),
            '-DENABLE_MPI=%s' % ('+mpi' in self.spec),
        ]

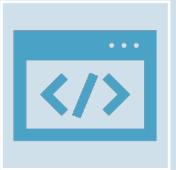
    def install(self, spec, prefix):
        mkdirp(prefix.bin)
        install('./spack-build/kripke', prefix.bin)
```



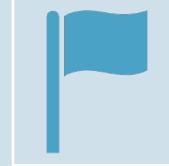
One package.py file per software project!

Source: Gregory Becker, Richarda Butler, Tamara Dahlgren, and Todd Gamblin. Managing HPC Software Complexity with Spack. RADIUSS Tutorial Series 2023. Virtual event. August 8-9, 2023.

What is a spack package (3)



DSL stand for domain specific language.



Provides directives [depends_on, version, patch, metadata]



Shell commands are to be invoked as Python function



Install methods easily call configure, make, make install as a Shell script would

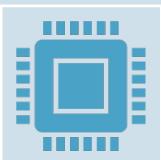
About other tools that automate builds



Conda/Mamba not ultimately targeted for HPC as they ship unoptimized binaries.



Containers are great and allow users to build environments but somebody has to build them!



Container performance, security or upgrade issues may slowdown its use on HPC.

Overview of the spack spec (1)

Spack provides a *spec* syntax to describe customized package configurations

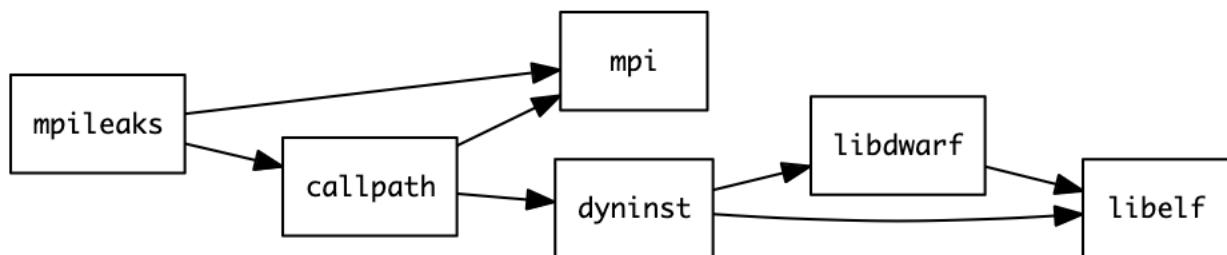
```
$ spack install mpileaks           unconstrained  
$ spack install mpileaks@3.3       @ custom version  
$ spack install mpileaks@3.3 %gcc@4.7.3    % custom compiler  
$ spack install mpileaks@3.3 %gcc@4.7.3 +threads  +/- build option  
$ spack install mpileaks@3.3 cppflags="-O3 -g3"  set compiler flags  
$ spack install mpileaks@3.3 target=cascadelake  set target microarchitecture  
$ spack install mpileaks@3.3 ^mpich@3.2 %gcc@4.9.3  ^ dependency constraints
```

- Each expression is a *spec* for a particular configuration
 - Each clause adds a constraint to the spec
 - Constraints are optional – specify only what you need.
 - Customize install on the command line!
- Spec syntax is recursive
 - Full control over the combinatorial build space

Overview of spack spec (2)

- MPileaks class has metadata, 2 direct dependencies (callpath & mpi).
- Thus, spack recursively inspects the class definitions for each dependency and construct a DAG representing those relationships
- It has a simplest possible spec, a single identifier (unconstrained):

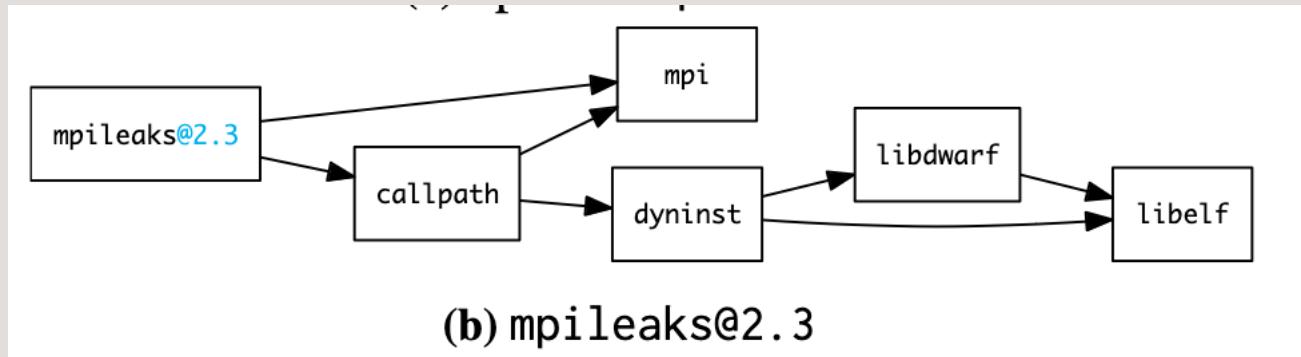
spack install mpileaks



(a) Spec for mpileaks

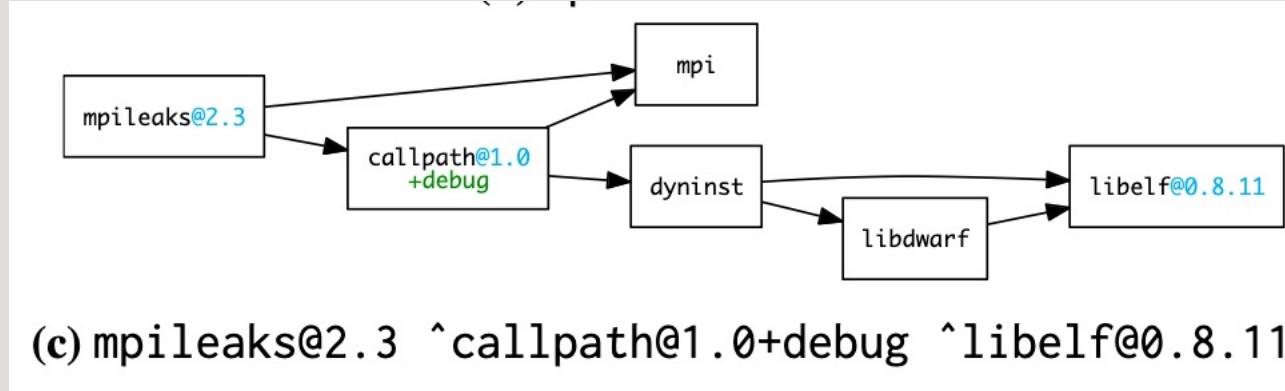
Overview of spack spec (3)

- To install mpileaks version 2.3
- ***spack install mpileaks@2.3***



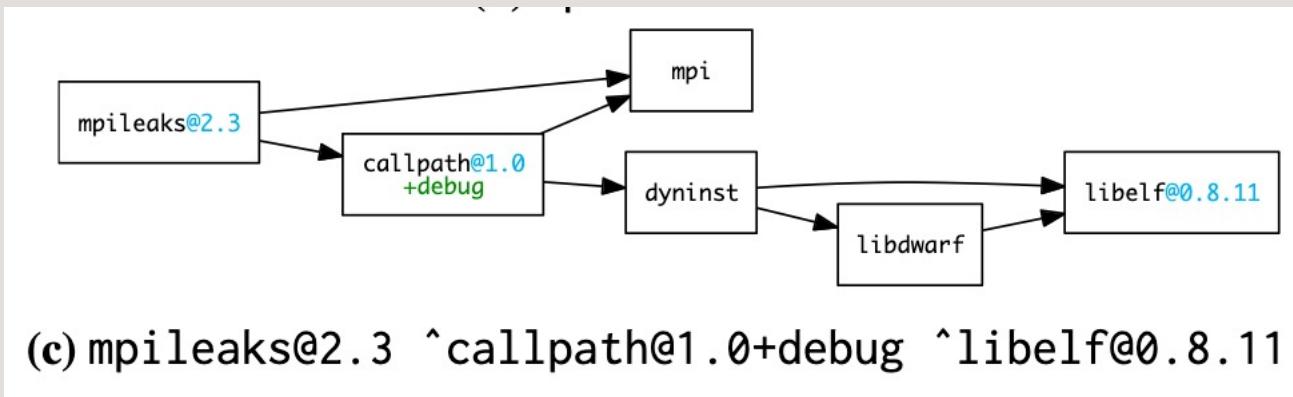
Overview of spack spec (4)

- To install mpileaks version 2.3 , which depends on callpath version 1.0, with debug build option, and which will also depend on libelf version 0.8.11.
- ***spack install mpileaks@2.3 ^callpath@1.0+debug ^libelf@0.8.11***



Overview of spack spec (5)

- If I wanted a particular minimum version I could have provided a range syntax with ":"
- ***spack install mpileaks@2.3:2.5.6***



Overview of spack concretization (1)

- Process that allows spack to achieve a concrete build
- User invokes “spack install” → causes a spec to be built

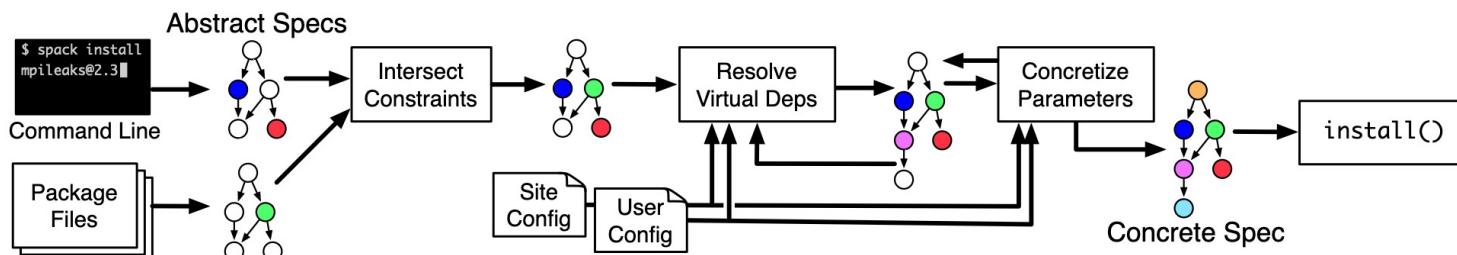


Figure 6: Spack’s concretization process.

Overview of spack concretization (2)

- Overview of a concretized tree

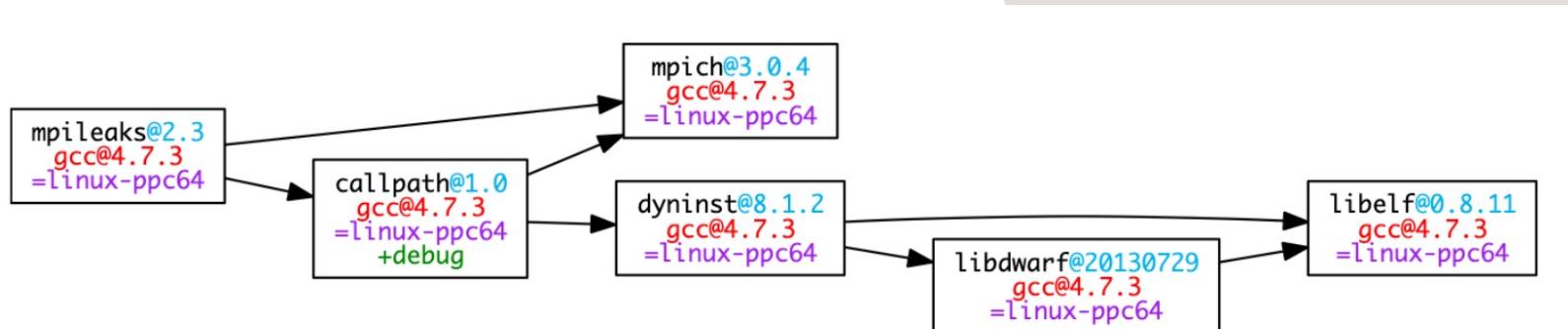
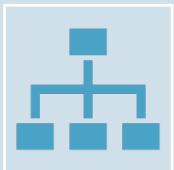


Figure 7: Concretized spec from Figure 2a.

Overview of spack concretization (2)



Spack is a package manager that can manage dependencies by ensuring consistent build.



Spack essentially is an ensemble for recipes

Spack mirrors (1)

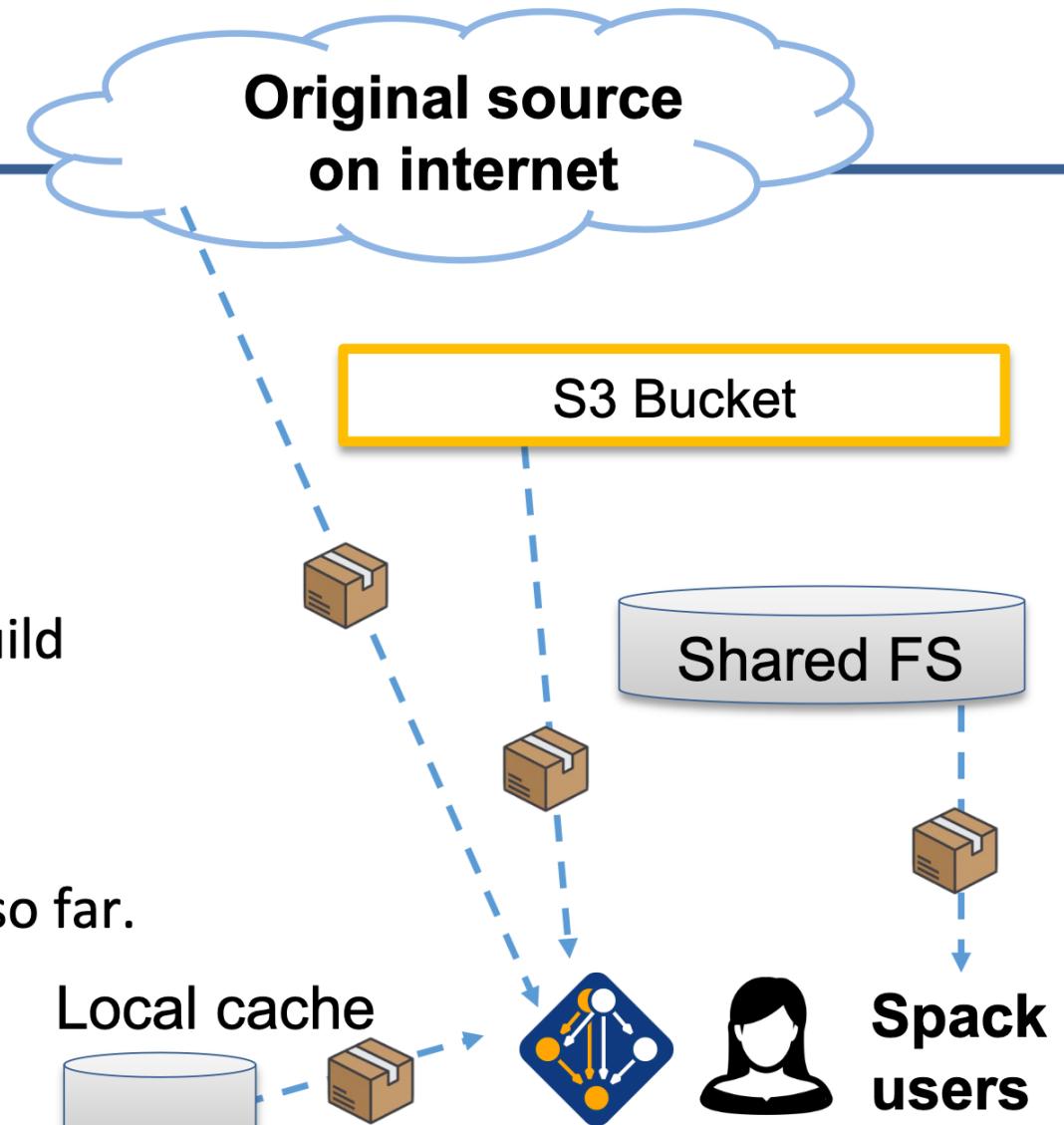
- Mirrors allow to accelerate installation time on a platform by shipping compatible binaries.
- Allow to easily share a spack environment (e.g. r, cuda etc ...).
- Allow to build packages either privately or publicly



Spack mirrors (2)

Spack mirrors

- Spack allows you to define *mirrors*:
 - Directories in the filesystem
 - On a web server
 - In an S3 bucket
- Mirrors are archives of fetched tarballs, repositories, and other resources needed to build
 - Can also contain binary packages
- By default, Spack maintains a mirror in `var/spack/cache` of everything you've fetched so far.
- You can host mirrors internal to your site
 - See the documentation for more details



Adding spack mirrors locally

```
[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$ spack mirror add v018 https://binaries.spack.io/releases/v0.18
[[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$ spack mirror add E4S https://cache.e4s.io/22.05
[[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$ spack mirror add spack-public https://mirror.spack.io
[[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$ spack buildcache keys --install --trust
==> Fetching https://cache.e4s.io/22.05/build_cache/_pgp/25645FA2B218FE55B4EF649E4345F04B40005581.pub
gpg: key 4345F04B40005581: public key "University of Oregon - E4S" imported
gpg: Total number processed: 1
gpg:          imported: 1
gpg: inserting ownertrust of 6
gpg: key A8E0CA3C1C2ADA2F: 7 signatures not checked due to missing keys
gpg: key A8E0CA3C1C2ADA2F: "Spack Project Official Binaries <maintainers@spack.io>" not changed
gpg: Total number processed: 1
gpg:          unchanged: 1
[[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$ spack mirror list
spack-public [sb] https://mirror.spack.io
E4S          [sb] https://cache.e4s.io/22.05
v018         [sb] https://binaries.spack.io/releases/v0.18
[kfotso@xsede.org@c3cpu-a5-u3-2 spack]$
```

Spack environments (1)

Making an environment allows you to focus on specific packages .

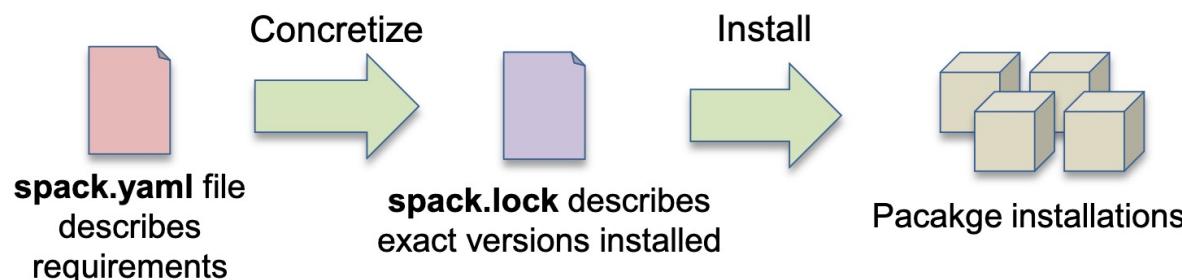
Very good approach for software reproducibility as it can be shared.

A user may create a container image out of an environment (**spack containerize**)

Better flexibility than Anaconda

Spack environments (2)

Spack environments enable users to build customized stacks from an abstract description



- spack.yaml describes project requirements
- spack.lock describes exactly what versions/configurations were installed, allows them to be reproduced.
- Can be used to maintain configuration of a software stack.
 - Can easily version an environment in a repository

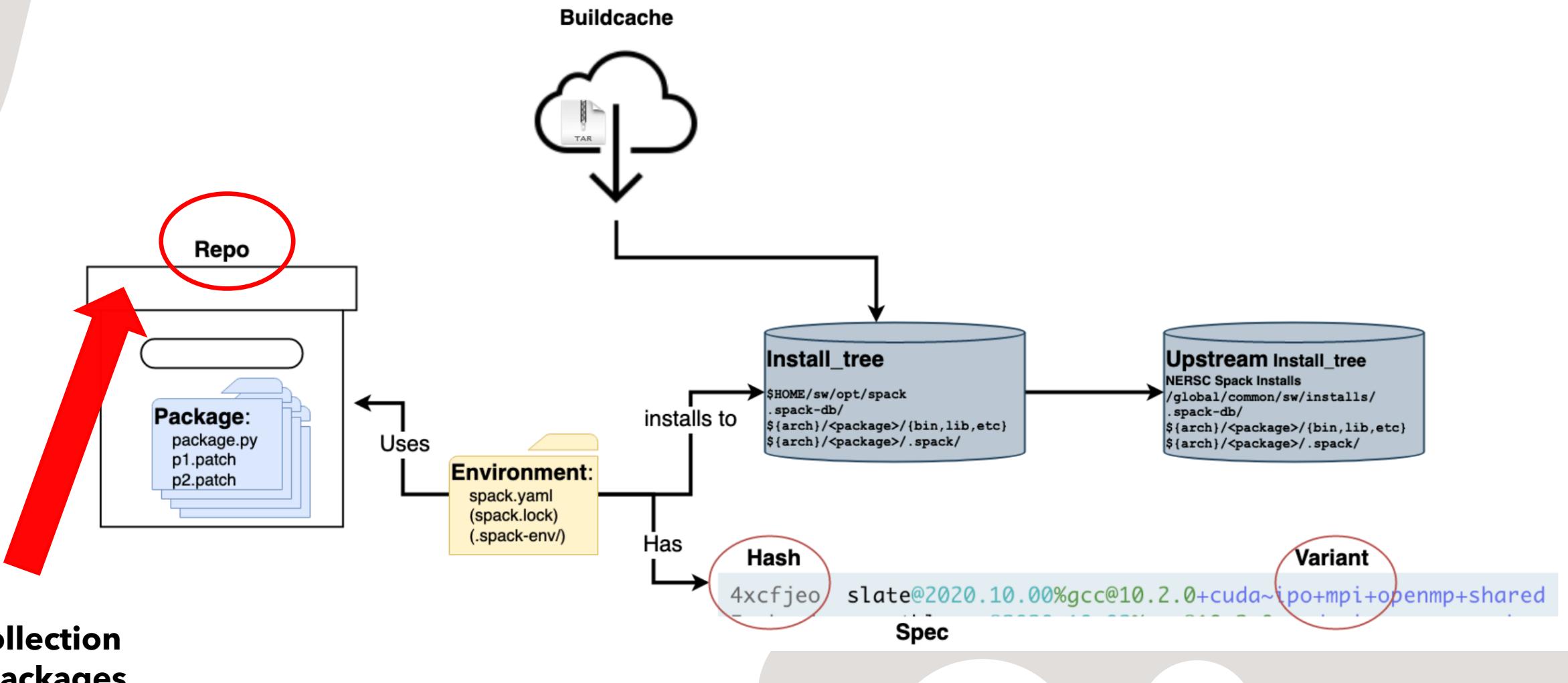
Simple spack.yaml file

```
spack:  
  # include external configuration  
  include:  
    - ./special-config-directory/  
    - ./config-file.yaml  
  
  # add package specs to the `specs` list  
  specs:  
    - hdf5  
    - libelf  
    - openmpi
```

Concrete spack.lock file (generated)

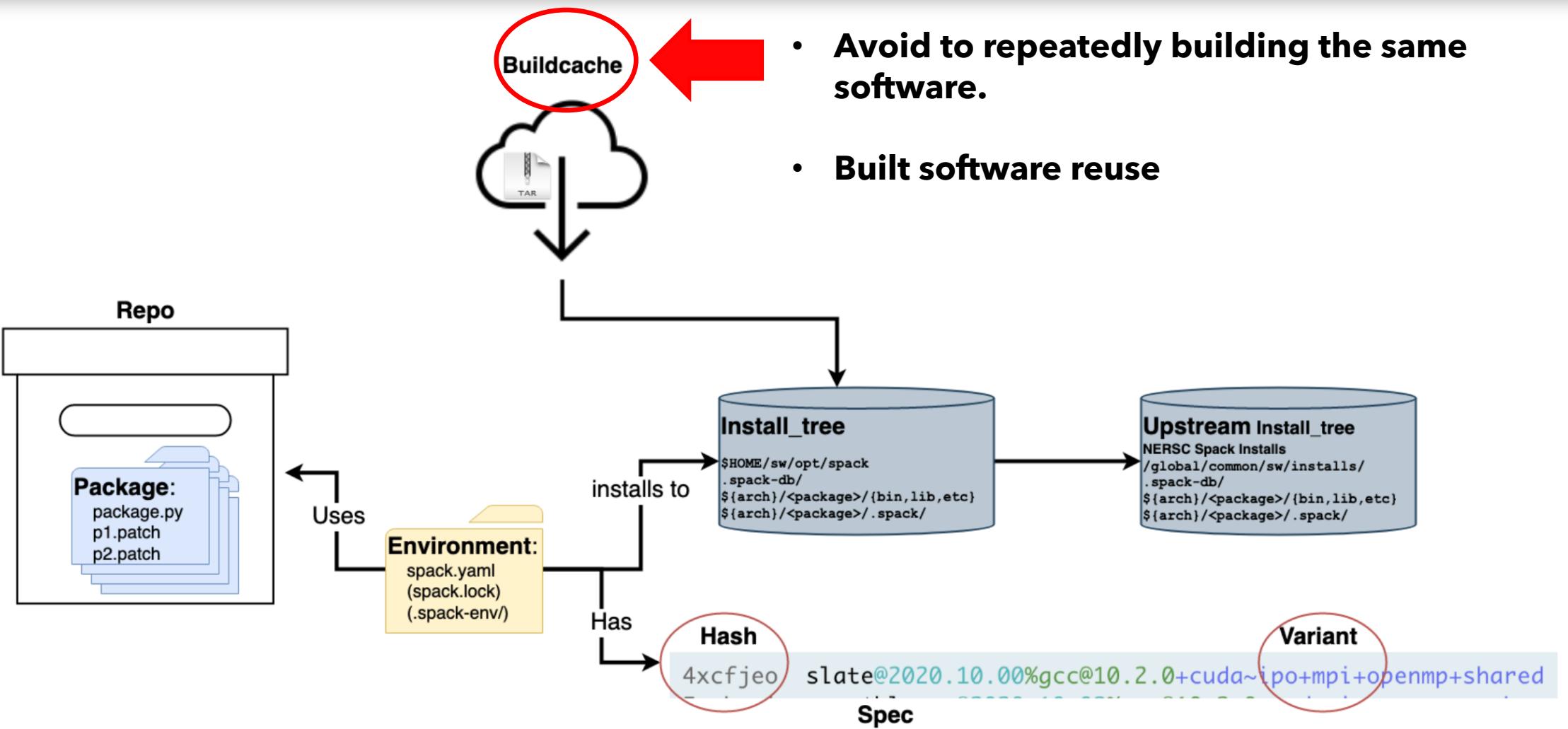
```
{  
  "concrete_specs": {  
    "6s63so2kstp3zyvjezglndmavy6l3nul": {  
      "hdf5": {  
        "version": "1.10.5",  
        "arch": {  
          "platform": "darwin",  
          "platform_os": "mojave",  
          "target": "x86_64"  
        },  
        "compiler": {  
          "name": "clang",  
          "version": "10.0.0-apple"  
        },  
        "namespace": "built-in",  
        "parameters": {}  
      }  
    }  
  }  
}
```

Overview of the installation process

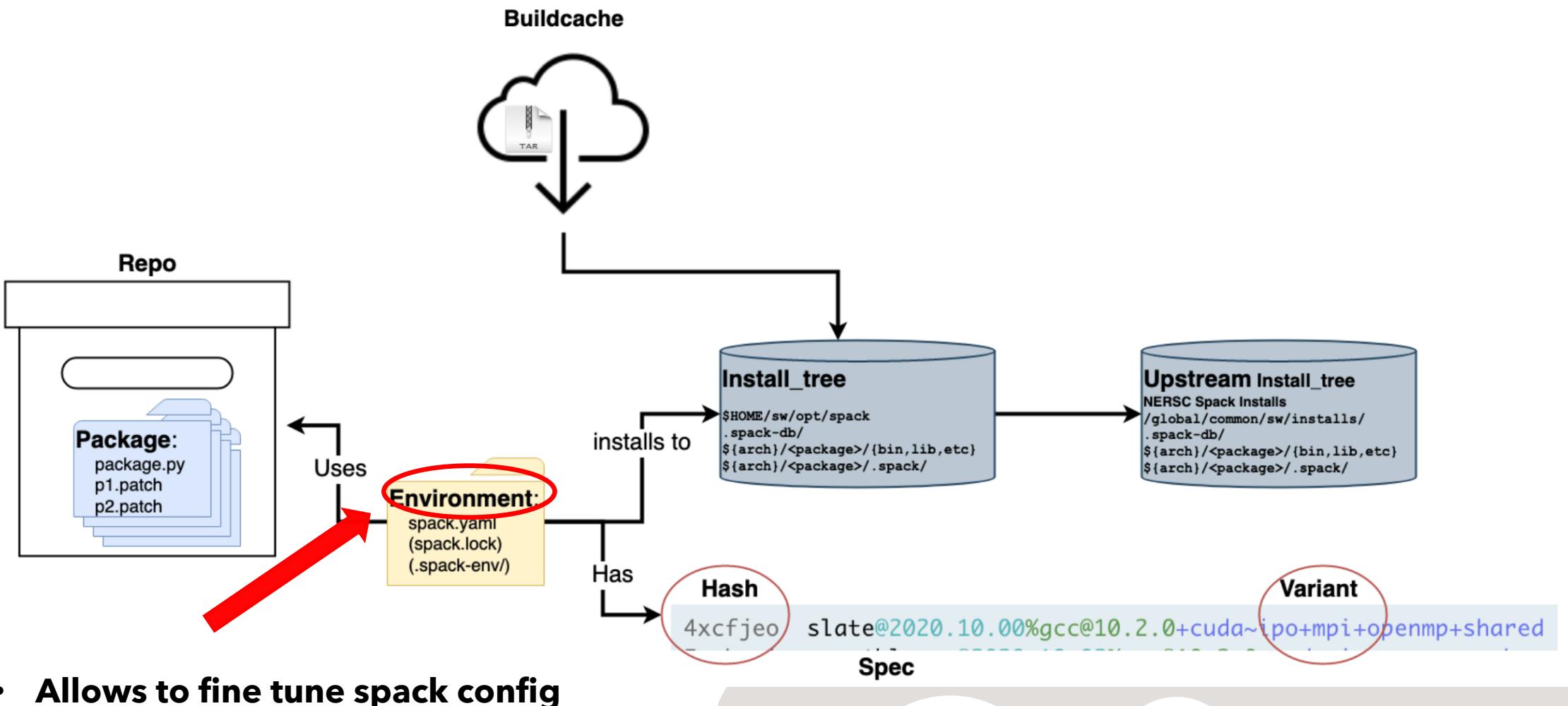


A collection
of packages

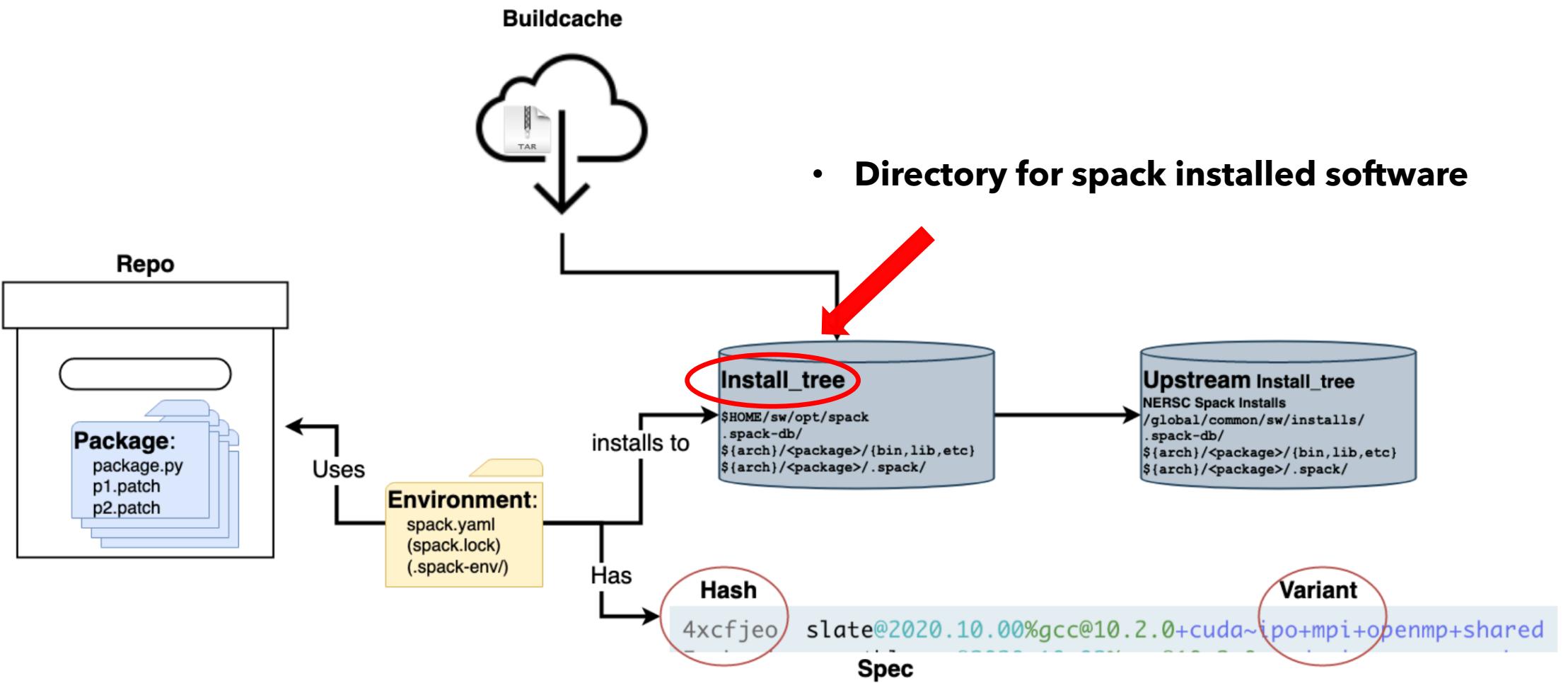
Overview of the installation process



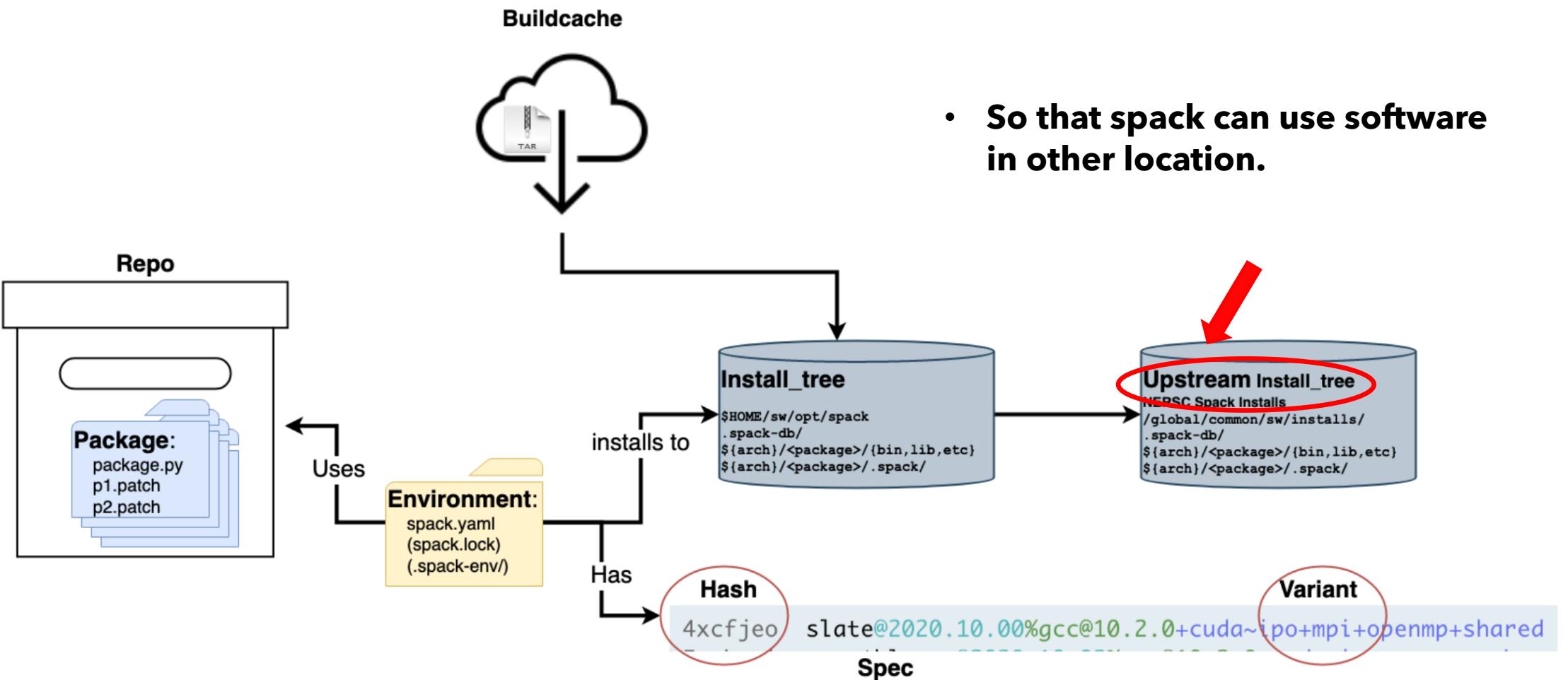
Overview of the installation process



Overview of the installation process



Overview of the installation process



Spack demo

← → ⌂

https://spack.readthedocs.io/en/latest/



Spack

latest

Search docs

BASICS

Feature Overview

Getting Started

Basic Usage

Spack for Homebrew/Conda Users

LINKS

Tutorial (spack-tutorial.rtfd.io)

Packages (packages.spack.io)

Binaries (binaries.spack.io)

REFERENCE

Configuration Files

Spack Settings (config.yaml)

Bootstrapping

Concretization Settings
(concretizer.yaml)

Package Settings (packages.yaml)



Official website containing tutorial:

https://spack.readthedocs.io/en/latest/

🏠 / Spack

Edit on GitHub

Spack

These are docs for the Spack package manager. For sphere packing, see [pyspack](#).

Spack is a package management tool designed to support multiple versions and configurations of software on a wide variety of platforms and environments. It was designed for large supercomputing centers, where many users and application teams share common installations of software on clusters with exotic architectures, using libraries that do not have a standard ABI. Spack is non-destructive: installing a new version does not break existing installations, so many configurations can coexist on the same system.

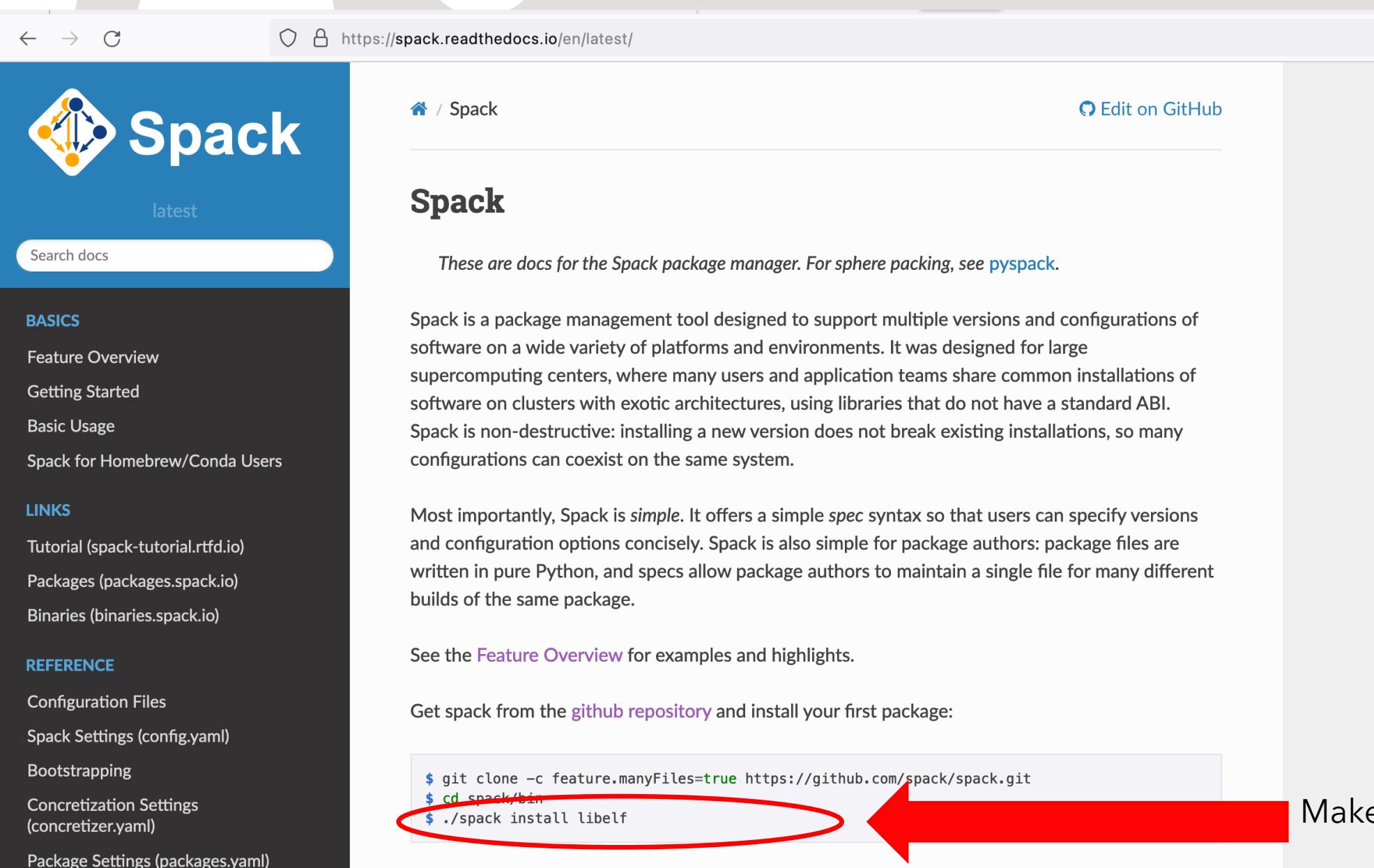
Most importantly, Spack is *simple*. It offers a simple `spec` syntax so that users can specify versions and configuration options concisely. Spack is also simple for package authors: package files are written in pure Python, and specs allow package authors to maintain a single file for many different builds of the same package.

See the [Feature Overview](#) for examples and highlights.

Get spack from the [github repository](#) and install your first package:

```
$ git clone -c feature.manyFiles=true https://github.com/spack/spack.git
$ cd spack/bin
$ ./spack install libelf
```

Spack demo (1)



The screenshot shows the Spack documentation homepage at <https://spack.readthedocs.io/en/latest/>. The page features a blue header with the Spack logo and navigation links for 'latest' and 'Search docs'. The main content area has a dark background with white text. It starts with a brief introduction: "These are docs for the Spack package manager. For sphere packing, see [pyspack](#)". Below this, there's a detailed paragraph about Spack's design and capabilities. Further down, under 'LINKS', there are links to a tutorial, packages, and binaries. Under 'REFERENCE', there are links to configuration files, bootstrapping, concretization settings, and package settings. A red arrow points from the text "Make sure to install with libelf!!!" to the command line example in the code block.

latest

Search docs

BASICS

- Feature Overview
- Getting Started
- Basic Usage
- Spack for Homebrew/Conda Users

LINKS

- Tutorial ([spack-tutorial.rtfd.io](#))
- Packages ([packages.spack.io](#))
- Binaries ([binaries.spack.io](#))

REFERENCE

- Configuration Files
- Spack Settings ([config.yaml](#))
- Bootstrapping
- Concretization Settings ([concretizer.yaml](#))
- Package Settings ([packages.yaml](#))

These are docs for the Spack package manager. For sphere packing, see [pyspack](#).

Spack is a package management tool designed to support multiple versions and configurations of software on a wide variety of platforms and environments. It was designed for large supercomputing centers, where many users and application teams share common installations of software on clusters with exotic architectures, using libraries that do not have a standard ABI. Spack is non-destructive: installing a new version does not break existing installations, so many configurations can coexist on the same system.

Most importantly, Spack is *simple*. It offers a simple `spec` syntax so that users can specify versions and configuration options concisely. Spack is also simple for package authors: package files are written in pure Python, and specs allow package authors to maintain a single file for many different builds of the same package.

See the [Feature Overview](#) for examples and highlights.

Get spack from the [github repository](#) and install your first package:

```
$ git clone -c feature.manyFiles=true https://github.com/spack/spack.git
$ cd spack/bin
$ ./spack install libelf
```

Make sure to install with libelf!!!

Spack demo (2)

```
$ acompile --ntasks=4 --time=00:30:00
```

Acompile is for installing packages

```
[kfotso@xsede.org@c3cpu-a5-u3-2 software]$ export TMP=/scratch/alpine/$USER  
[kfotso@xsede.org@c3cpu-a5-u3-2 software]$ export TEMP=/scratch/alpine/$USER  
[kfotso@xsede.org@c3cpu-a5-u3-2 software]$ export TMPDIR=/scratch/alpine/$USER  
[kfotso@xsede.org@c3cpu-a5-u3-2 software]$ export TEMPDIR=/scratch/alpine/$USER
```

To be on the safe side and not fill /tmp

Packages contained in spack

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack list
```

3dtk	libxcb
3proxy	libcomposite
7zip	libxcrypt
abacus	libxcursor
abduco	libxdamage
abi-compliance-checker	libxdmcp
abi-dumper	libxevie
abinit	libxext
abseil-cpp	libxfixed
abyss	libxfont
accfft	libxfont2
acct	libxfontcache
accumulo	libxft
ace	libxi
acfl	libxinerama
ack	libxkbcommon
acl	libxkbfile
acpica-tools	libxkbui
acpid	libxml2
activeharmony	libxmu
activemq	libxyn

py-gql	r-dbplyr
py-grandalf	r-debugme
py-grapheme	r-decipher
py-graphlib-backport	r-delayedarray
py-graphql-core	r-delayedmatrixstats
py-graphviz	r-deldir
py-gravity	r-dendextend
py-greenlet	r-densvis
py-requests	r-deoptim
py-griddataformats	r-deoptimr
py-griffe	r-deriv
py-grpcio	r-desc
py-grpcio-tools	r-deseq
py-gsd	r-deseq2
py-gssapi	r-desolve
py-gsutil	r-devtools
py-gtdbtk	r-dexseq
py-gunicorn	r-diagram
py-gxformat2	r-diagrammer
py-gym	r-dicekriging
nv-h11	r-dichromat

7334 packages!

Fine search of packages in spack (wildcard)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack list 'r-*'
```

```
r-a4
r-a4base
r-a4classif
r-a4core
r-a4preproc
r-a4reporting
r-abadata
r-abaenrichment
r-abind
r-absseq
r-acde
r-acepack
r-acgh
r-acme
r-ada
r-adabag
r-ade4
r-adegenet
r-adegraphics
r-a4
r-dbi
r-dbplyr
r-debugme
r-decipher
r-delayedarray
r-delayedmatrixstats
r-deldir
r-dendextend
r-densvis
r-deoptim
r-deoptimr
r-deriv
r-desc
r-deseq
r-deseq2
r-desolve
r-devtools
r-dexseq
r-diagram
r-hdf5i
r-hdo-db
r-hdrconde
r-here
r-hexbin
r-hh
r-highr
r-hmisc
r-hms
r-hoardr
r-htmltable
r-htmltools
r-htmlwidgets
r-htppcode
r-htppuv
r-httr
r-httr2
r-hwriter
r-hydrogof
```

```
r-paleotree
r-pamr
r-pan
r-parallely
r-parallelmap
r-paramhelpers
r-party
r-partykit
r-patchwork
r-pathview
r-pbapply
r-pbdzmq
r-pbivnorm
r-pbkrtest
r-pcamethods
r-pcapp
r-pegas
r-permute
r-pfam-db
r-ruv
r-rvcheck
r-rversions
r-rvest
r-rviennacl
r-rzmq
r-s2
r-s4vectors
r-samr
r-sandwich
r-sass
r-satellite
r-scaledmatrix
r-scales
r-scater
r-scattermore
r-scatterpie
r-scatterplot3d
r-scdblfinder
```

1132 r packages!

Spack install(1)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack install zlib  
[+] /projects/kefo9343/software/spack/opt/spack/linux-rhel8-zen/gcc-8.4.1/zlib-1.2.13-axwtx3dzaqi47indh2blq72sxuqgexq  
[kefo9343@c3cpu-c11-u3-1 software]$ spack spec zlib  
Input spec  
-----  
- zlib  
  
Concretized  
-----  
[+] zlib@1.2.13%gcc@8.4.1+optimize+pic+shared build_system=makefile arch=linux-rhel8-zen
```

Zlib was already installed

Concretized Specs for the installed

Spack install(2)

```
kefo9343@c3cpu-c11-u3-1 software]$ spack find zlib  
-- linux-rhel8-zen / gcc@8.4.1 -----  
zlib@1.2.13  
=> 1 installed package
```

To find all the installed
Zlib on my system

Spack install (3)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack info zlib
```

MakefilePackage: zlib

Description:

A free, general-purpose, legally unencumbered lossless data-compression library.

Homepage: <https://zlib.net>

Preferred version:

1.2.13 <http://zlib.net/fossils/zlib-1.2.13.tar.gz>



Most stable version

Safe versions:

1.2.13 <http://zlib.net/fossils/zlib-1.2.13.tar.gz>

Deprecated versions:

1.2.12 <http://zlib.net/fossils/zlib-1.2.12.tar.gz>

Spack install (4)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack install zlib@1.2.12
--> Warning: using "NodeArgument(id='0', pkg='zlib')@1.2.12" which is a deprecated version
--> Installing zlib-1.2.12-cc7ywoh7xk53osrb5r2hufhbishy3wps [1/1]
--> No binary for zlib-1.2.12-cc7ywoh7xk53osrb5r2hufhbishy3wps found: installing from source
--> Warning: zlib@1.2.12 is deprecated and may be removed in a future Spack release.
--> Fetch anyway? [y/N] y
--> Fetching https://mirror.spack.io/_source-cache/archive/91/91844808532e5ce316b3c010929493c0244f3d37593af6de04f71821d5136d9.tar.gz
--> Applied patch /projects/kefo9343/software/spack/var/spack/repos/builtin/packages/zlib/configure-cc.patch
--> zlib: Executing phase: 'edit'
--> zlib: Executing phase: 'build'
--> zlib: Executing phase: 'install'
--> zlib: Successfully installed zlib-1.2.12-cc7ywoh7xk53osrb5r2hufhbishy3wps
Stage: 11.16s. Edit: 0.41s. Build: 0.92s. Install: 1.07s. Post-install: 0.10s. Total: 13.76s
[+] /projects/kefo9343/software/spack/opt/spack/linux-rhel8-zen/gcc-8.4.1/zlib-1.2.12-cc7ywoh7xk53osrb5r2hufhbishy3wps
```



Deprecated version

Spack install (5)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack find zlib  
-- linux-rhel8-zen / gcc@8.4.1 -----  
zlib@1.2.12  zlib@1.2.13  
==> 2 installed packages
```



Spack install (6)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack versions zlib
==> Safe versions (already checksummed):
  1.2.13  1.2.12  1.2.11  1.2.8  1.2.3
==> Remote versions (not yet checksummed):
  1.3      1.2.7.2   1.2.6    1.2.5    1.2.4.2   1.2.3.8   1.2.3.4   1.2.2.4   1.2.2    1.2.0.8   1.2.0.4   1.2.0    1.1.1    1.0.7    1.0.2    0.95     0.91     0.8
  1.2.10   1.2.7.1   1.2.5.3   1.2.4.5   1.2.4.1   1.2.3.7   1.2.3.3   1.2.2.3   1.2.1.2   1.2.0.7   1.2.0.3   1.1.4    1.1.0    1.0.6    1.0.1    0.94     0.79
  1.2.9    1.2.7    1.2.5.2   1.2.4.4   1.2.4    1.2.3.6   1.2.3.2   1.2.2.2   1.2.1.1   1.2.0.6   1.2.0.2   1.1.3    1.0.9    1.0.5    1.0-pre   0.93     0.71
  1.2.7.3  1.2.6.1   1.2.5.1   1.2.4.3   1.2.3.9   1.2.3.5   1.2.3.1   1.2.2.1   1.2.1    1.2.0.5   1.2.0.1   1.1.2    1.0.8    1.0.4    0.99     0.92     0.9
[kefo9343@c3cpu-c11-u3-1 software]$
```

Spack install (7)

We install tcl without any spec

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack install tcl
[+] /projects/kefo9343/software/spack/opt/spack/linux-rhel8-zen/gcc-8.4.1/zlib-ng-2.1.3-rviugiapkgwvgpo73bkww4yktl2jr4lv
==> Installing tcl-8.6.12-bah3oyl7byevonhftmzcjp3mr371 [2/2]
==> No binary for tcl-8.6.12-bah3oyl7byevonhftmzcjp3mr371 found: installing from source
==> Fetching https://mirror.spack.io/_source-cache/archive/26/26c995dd0f167e48b11961d891ee555f680c175f7173ff8cb829f4ebcde4c1a6.tar.gz
==> No patches needed for tcl
==> tcl: Executing phase: 'autoreconf'
==> tcl: Executing phase: 'configure'
==> tcl: Executing phase: 'build'
==> tcl: Executing phase: 'install'
==> tcl: Successfully installed tcl-8.6.12-bah3oyl7byevonhftmzcjp3mr371
  Stage: 3.29s.  Autoreconf: 0.00s.  Configure: 4.51s.  Build: 42.97s.  Install: 24.86s.  Post-install: 7.36s.  Total: 1m 23.16s
```

Spack install (8)

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack install tcl ^zlib@1.2.8
==> Warning: using "NodeArgument(id='0', pkg='zlib')@1.2.8" which is a deprecated version
==> Installing zlib-1.2.8-p7udpbni6vjyvmgnip4jo20ouq5bw6l7 [1/2]
==> No binary for zlib-1.2.8-p7udpbni6vjyvmgnip4jo20ouq5bw6l7 found: installing from source
==> Warning: zlib@1.2.8 is deprecated and may be removed in a future Spack release.
==> Fetch anyway? [y/N] y
==> Using cached archive: /projects/kefo9343/software/spack/var/spack/cache/_source-cache/archive/36/36658cb768a54c1d4dec43c3116c27ed89
==> No patches needed for zlib
==> zlib: Executing phase: 'edit'
==> zlib: Executing phase: 'build'
==> zlib: Executing phase: 'install'
==> zlib: Successfully installed zlib-1.2.8-p7udpbni6vjyvmgnip4jo20ouq5bw6l7
  Stage: 2.82s. Edit: 0.29s. Build: 0.93s. Install: 0.13s. Post-install: 0.10s. Total: 4.37s
[+] /projects/kefo9343/software/spack/opt/spack/linux-rhel8-zen/gcc-8.4.1/zlib-1.2.8-p7udpbni6vjyvmgnip4jo20ouq5bw6l7
==> Installing tcl-8.6.12-xlkosam6hm56w4qeosdylnmodtzhoduq [2/2]
==> No binary for tcl-8.6.12-xlkosam6hm56w4qeosdylnmodtzhoduq found: installing from source
==> Using cached archive: /projects/kefo9343/software/spack/var/spack/cache/_source-cache/archive/26/26c995dd0f167e48b11961d891ee555f68
==> No patches needed for tcl
==> tcl: Executing phase: 'autoreconf'
==> tcl: Executing phase: 'configure'
==> tcl: Executing phase: 'build'
==> tcl: Executing phase: 'install'
==> tcl: Successfully installed tcl-8.6.12-xlkosam6hm56w4qeosdylnmodtzhoduq
  Stage: 0.97s. Autoreconf: 0.00s. Configure: 4.42s. Build: 42.94s. Install: 23.48s. Post-install: 9.17s. Total: 1m 21.15s
[+] /projects/kefo9343/software/spack/opt/spack/linux-rhel8-zen/gcc-8.4.1/tcl-8.6.12-xlkosam6hm56w4qeosdylnmodtzhoduq
```

We add the spec

Spack graph

```
[kefo9343@c3cpu-c11-u3-1 software]$ spack graph tcl
o tcl@8.6.12/bah3oyl
o zlib-ng@2.1.3/rviugia
```

The background features a minimalist design with several large, semi-transparent circles in light gray and white. These circles overlap each other in a fluid, organic pattern across the entire frame.

Questions?