



Spack

A Package Manager For Supercomputers

Moritz Siegel

Why A Package Manager For Supercomputers?

Many installations of the **same** software but with different:

- ▶ Hardware
- ▶ Versions
- ▶ Compilers
- ▶ Flags
- ▶ Dependencies

Example MPI:

18 providers x 30 versions x 6 compilers x 22 flags x 31 dependencies =
= **2 Million variants** on VSC-5 alone!

Three Spack Trees



VSC-4
Intel
CPU
skylake



VSC-5
AMD
CPU
zen



VSC-5
AMD
GPU
cuda-zen

The current spack tree is shown on the left of the command line **prompt**:

```
1 zen trainee00@l55:~$  
2 cuda-zen trainee00@l55:~$  
3 skylake trainee00@l44:~$
```

On login the **spack tree** for the current architecture is set.

Packages from one tree do not work on other architecture!

All spack commands work on the **current tree**.

Type **skylake**, **zen** or **cuda-zen** to **switch** to a spack tree:

```
1 zen trainee00@155:~$ cuda-zen
2 cuda-zen trainee00@155:~$ skylake
3 skylake trainee00@155:~$ zen
4 zen trainee00@155:~$
```

You are on the **same node 155** the whole time.

Find Your Package

Find your package, like **openmpi**:

```
1 zen trainee00@l55:~$ spack find openmpi
```

Use "@" to show specific **version** only:

```
2 zen trainee00@l55:~$ spack find openmpi@3
```

```
3 zen trainee00@l55:~$ spack find openmpi@4.1.4
```

All **python** packages have a leading "py-" like in **py-numpy**.

All **R** packages start with "r-" like in **r-brew**.

Add `-l` to show the unique package **hash** too:

```
1 zen trainee00@l55:~$ spack find -l openmpi@4.1.4
2 -- linux-almalinux8-zen / gcc@8.5.0 -----
3 3yligjt openmpi@4.1.4
4 ...
```

Use **any** spack command with this hash, and a "/" in front:

```
2 zen trainee00@l55:~$ spack find -l /3yligjt
3 -- linux-almalinux8-zen / gcc@8.5.0 -----
4 3yligjt openmpi@4.1.4
5 ==> 1 installed package
```

Use `spack search` to search for a **hash** of an old VSC installation:

```
1 zen trainee00@l155:~$ spack search /asdc2mk
2
3 ==> The hash asdc2mkqlgtelulq5nu6ctd4b5stefag refers to python from vsc4 w
4     python +bz2+ctypes+dbm~debug+libxml2+lzma~nis~optimizations+pic+pyexpa
5
6 ==> Similar python packages in installation zen:
7 -- linux-almalinux8-zen / gcc@8.5.0 -----
8 ...
```

You get some similar modules on the **current** spack tree.

Use "%" to only show packages **compiled** with intel, gcc, aocc, etc:

```
1 skylake trainee00@l44:~$ spack find openmpi %intel
2 zen trainee00@l55:~$ spack find openmpi %aocc
3 cuda-zen trainee00@l55:~$ spack find openmpi %gcc
```

Combine "%" and "@" to show specific **compiler version** only:

```
4 zen trainee00@l55:~$ spack find openmpi %gcc@9
```

Use `spack find -v` to see the **flags** of a package:

```
1 zen trainee00@l55:~$ spack find -v openmpi@4.1.4
```

Use "+" or "~" to show only packages that **have/not have** that flag:

```
2 cuda-zen trainee00@l55:~$ spack find openmpi +cuda
```

Use `spack find -d` to show the **dependencies** of a package:

```
1 zen trainee00@l155:~$ spack find -d /3yligjt
```

Use "`^`" to only show packages **A** that depend on **B**:

```
2 skylake trainee00@l144:~$ spack find -d py-numpy ^intel-oneapi-mkl
```

```
3 zen trainee00@l155:~$ spack find -d py-numpy ^openblas
```

Load Your Package

Use `spack load` to load your package:

```
1 zen trainee00@l155:~$ spack load openmpi@4.1.4
```

Add the unique **hash** to load that **exact** package:

```
2 zen trainee00@l155:~$ spack load py-numpy/4qshzxi  
3 cuda-zen trainee00@l155:~$ spack load py-numpy/bctcmkr  
4 skylake trainee00@l144:~$ spack load py-numpy/b3lgcbu
```

All these packages are python's **numpy**, but built for different architectures.

List & Unload

Type `spack find --loaded` to list loaded packages:

```
1 zen trainee00@l55:~$ spack find --loaded
2 -- linux-almalinux8-zen3 / gcc@12.2.0 -----
3 o6ui7qb berkeley-db@18.1.40          4o2lz14 libbsd@0.11.5
4 qta2z6t openssl@1.1.1s              sultos7 py-setuptools@63.0.0
5 ...
```

Unload a package with `spack unload mypackage`:

```
1 zen trainee00@l55:~$ spack unload py-numpy/4qshzxi
```

Type `spack unload` to unload **all** loaded packages.

Every package installed with spack creates a module file:

```
1 <package>/<version>-<compiler>-<version>-<hash>  
2 openmpi/3.1.6-aocc-4.0.0-k4glowk  
3 openmpi/4.1.4-gcc-12.2.0-je3hmfk  
4 openmpi/4.1.4-intel-2021.7.1-lex74to
```

The **hash** at the end, like **k4glowk** is the same as in spack.

- Login to VSC-4/5.
- Switch between the three spack trees **zen**, **skylake** and **cuda-zen**.
- Find a **hdf5** variant compiled with **aocc** at **zen**.
- Find a **openmpi** with **cuda** support at **cuda-zen**.
- Find a **numpy** including **intel-oneapi-mkl** at **skylake**.
- Load any **openmpi** package, then list all the loaded packages.
- Find out what old package once was `/asdc2mk`.