Installing Software on Alpine with **Spack**

Research Computing
UNIVERSITY OF COLORADO BOULDER

Installing software on Alpine with Spack

Date: February 20, 2024

Instructor: Trevor Hall

Format: Primer

- Website: www.rc.colorado.edu
- Documentation: https://curc.readthedocs.io
- Helpdesk: rc-help@colorado.edu
- Survey: http://tinyurl.com/curc-survey18

Contributors: Andy Monaghan, Brandon Reyes, Layla Freeborn

Slides

https://github.com/ResearchComputing/alpine_spack_primer

Session Overview

Introduction

- Installing software on Alpine
- Description of Spack

Setting up Spack on Alpine

- Logging in
- Using Spack for the first time
- Starting an interactive session and activating Spack

Installing Software with Spack

- Loading Spack in your HPC Job
- Useful Spack Commands

Using Spack Virtual Environments

- Creating and accessing virtual environments

Strategies for more complex installations

- Legacy Software
- Compilers
- Complex Virtual Environments

Installing Software on Alpine

- There are numerous ways to install software on Alpine:
  - Use pre-compiled binaries
  - Compile from source
  - Within virtual environments (via Conda, Miniconda, or Mamba)
  - **Using a package manager for HPC systems (Spack)**
  - Using containers (Apptainer/Singularity)

Additional information:
https://github.com/ResearchComputing/research-software-curc

Installing Software on Alpine

- Definitions
  - **Building-** a generic term describing the overall installation process that includes compiling
  - **Compiling**- the process of converting source code to an executable
  - **Linking**- the process of combining pieces of code and data into a single file that can be loaded into memory and executed
  - **Installing**- any process that results in executables

Challenges of Installing Software from Source

Challenges:

- Varies heavily from software to software
- Often poorly documented
- Sometimes requires many tools and dependencies
- Depends heavily on how well the software is built
- May not be compatible with your system

… in all, source installations can be challenging to complete.

Simplifying Installations with Spack

- How can we simplify source installations?
  - **Package Managers** – Tools that automate installing, maintaining, and configuring software and any dependencies
  - **Environments** – A collection of resources that are available in a self-contained 'bubble'

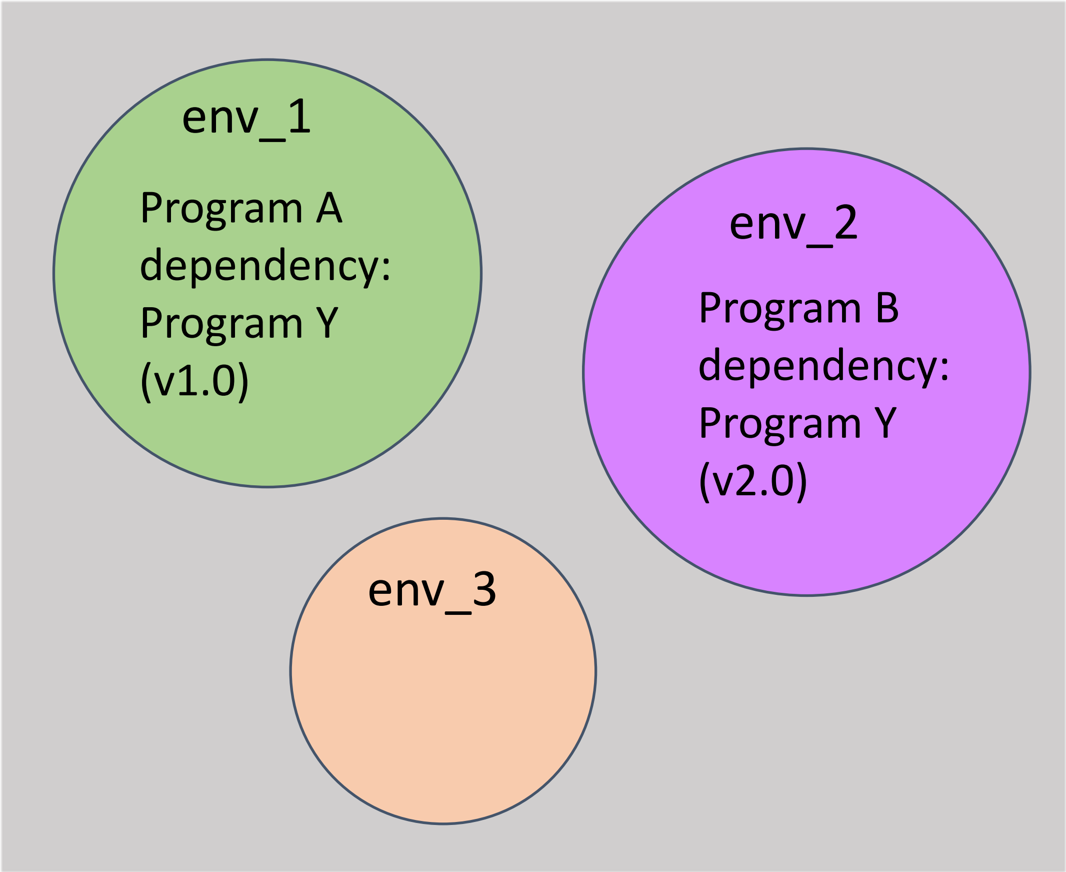
Simplifying Installations with Spack

Think of virtual environments as self-contained bubbles.

env_1 contains all the dependencies of ‘Program A’.

env_2 contains all the dependencies of ‘Program B’.

The environments do not interact.


env_1
Program A dependency: Program Y (v1.0)
env_2
Program B dependency: Program Y (v2.0)
env_3

Simplifying Installations with Spack

Your workflow requires two programs, ‘Program A’ and ‘Program B’.

- ‘Program A’ depends on ‘Program Y’ **v1.0**
- ‘Program B’ depends on ‘Program Y’ **v2.0**

What do you do?!

Logging into CU Research Computing

login to CURC via your terminal:

```
ssh <username>@login.rc.colorado.edu
```

…or login to CURC via your browser:

https://ondemand-rmacc.rc.colorado.edu

(once logged in, navigate to **Clusters -> Alpine shell**)

Additional information:
https://curc.readthedocs.io/en/latest/access/logging-in.html
https://curc.readthedocs.io/en/latest/gateways/OnDemand.html

Simplifying Installations with Spack

- Spack is available as a module, which can easily be loaded from a compute node

```
[user@login11 ~]$ acompile --help
[user@login11 ~]$ acompile --ntasks=4
…
[user@c3cpu-a5-u28-1 ~]$ module load spack
```

Note: when you log in to CURC, you'll be on a ***login*** node. You'll need to be on a ***compute*** node to use Spack. The **acompile** command allows you to quickly start an interactive job on a compute node.

Simplifying Installations with Spack

- Environments are created and programs are installed in a few simple steps

```
[user@c3cpu-a5-u28-1 ~]$ module load spack
[user@c3cpu-a5-u28-1 ~]$ spack env create my_first_env
[user@c3cpu-a5-u28-1 ~]$ spacktivate my_first_env
```

*Don’t install packages outside of an environment!**

Simplifying Installations with Spack

- Packages are installed within **activated** environments

Using `spack install`:

```
spack install --add fastqc             #install default fastqc
spack install --add fastqc@0.11.9      #install specific version of fastqc
spack find                             #view packages installed in environment
```

Simplifying Installations with Spack

- **Spack installations can be slow** but will progress more quickly with more cores.
- Spack builds all packages **in parallel.**
  - The default parallelism is equal to the number of cores available to the process, up to 16.

Simplifying Installations with Spack

- Additional useful Spack commands

```
spack env list                     # list all your environments
spack remove <env>                 # remove an environment
spack uninstall <packagename>      # remove package
spack env status                   # check which env you’re in
spack info <packagename>           # prints detailed package info
spack find                         # show installed packages
despacktivate                      # deactivate environment
spack spec <packagename>           # list packages pla
```

Simplifying Installations with Spack

- Useful spack file paths

```
# root of the spack install tree:
/projects/$USER/software/spack


# location of package executables - these are symbolically linked to
the installation tree subdirectory:
/projects/$USER/spack/environments/<env>/.spack-env/view/bin


# location of spack config file:
/home/$USER/.spack/config.yaml
```

Simplifying Installations with Spack

Hands-on exercise

Objectives:

1) Create a Spack environment
2) Install fastqc in your Spack environment

Estimated time to complete: 10 minutes

Running Alpine batch jobs with Spack

```
[user@login11 ~]$ nano runspack.sh  #Step 1: open new job script in editor
```

```
#!/bin/bash
# job script name: runspack.sh

#SBATCH --partition=amilan
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00

module purge
module load spack
spack activate my_first_env
./myprogram.exe
```

```
# Step 2: Write job script
```

https://curc.readthedocs.io/en/latest/running-jobs/batch-jobs.html

```
[user@login11 ~]$ sbatch runspack.sh  #Step 3: Schedule job
```

Using Spack to install legacy software

How can I use Spack to install older software?

- If unspecified, Spack will install and use the default version of a package (usually the newest stable version)
- If you need an older version, you can specify with the ‘@’ operator
  - Note that you may need older compilers or dependencies as well

```
[user@c3cpu-a5-u28-1 ~]$ spack info fastqc                         #Find information about a specific package, including versions

Package:   A quality control tool for high throughput sequence data.

Homepage: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Preferred version: 0.11.9  https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip

Safe Versions: 0.11.7  https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip

[...]

[user@c3cpu-a5-u28-1 ~]$ spack install fastqc@0.11.7               #install the specific package with the ‘@’ operator
```

Using Spack to install Compilers

How can I use Spack to manage compilers?

- If unspecified, Spack will install and use a default compiler
- You can also install specific compilers in your environments!

```
spack install gcc@13.1.0                                  #install compiler outside of an environment
export gcc_location=$(spack location -I gcc@13.1.0)       #store compiler’s location in env variable
spack env create compiler_env                             #create new environment
spack env activate compiler_env                           #activate new environment
spack compiler remove gcc -a                              #remove any default compiler
spack compiler add $gcc_location                          #add new compiler to compiler list
spack install --add gcc@13.1.0                            #install the new compiler to environment
spack install --add fastqc%gcc@13.1.0                     #install software with the new compiler
```

Thank you!

Survey and feedback

http://tinyurl.com/curc-survey18