GWDG
AG Computing
Venkata Ayyalasomayajula

Introduction

In this exercise we will install **Lmod** and **spack** on your cluster. Then we use these tools to install additional software, provide environment modules and explore the various features involved.

Task 1: Prerequisites (5 min)

First, enable the EPEL repository to then be able to install the dependencies needed for *Lmod*.

```
$ sudo dnf install -y epel-release
$ sudo dnf install lua lua-devel lua-json lua-lpeg lua-posix lua-term lua-filesystem bc
$ sudo dnf install tcl tcl-devel
```

NOTE: The above packages (except the `-devel` versions) are required on the node image as well.

Install the dependencies needed for *spack*.

```
$ sudo dnf install curl findutils gcc-c++ gcc gcc-gfortran git gnupg2 hostname
$ sudo dnf install iproute lsb_release make patch
$ sudo dnf install python3 python3-pip python3-setuptools unzip
```

Create a new user account with limited access rights for the management of software.

```
$ sudo useradd -m swmanager
$ sudo mkdir /nfs/sw
$ sudo chown swmanager:swmanager /nfs/sw
```

Open a login shell for the new user account.

```
$ sudo su - swmanager
```

For the remainder of this exercise sheet, we always use the new user `swmanager` unless the command starts with `sudo`.

Hints

- If you want to test this setup on your cluster, make sure `/nfs` is a shared NFS mount.
- The user account `swmanager` has to be synced to the node images.
  `$ sudo wwctl overlay build`

Task 2: Install Lmod (5 min)

Download the source code for Lmod from the official repository:

```
$ wget https://github.com/TACC/Lmod/archive/refs/tags/9.4.5.tar.gz
```

Extract the tarball and install Lmod into `/nfs/sw/`

```
$ tar -xf 9.4.5.tar.gz
```

```
$ cd Lmod-9.4.5/
$ ./configure --prefix=/nfs/sw
$ make install
```

To make Lmod available to all users, create a symbolic link to the profile folder:
`$ sudo ln -s /nfs/sw/lmod/lmod/init/profile /etc/profile.d/modules.sh`

NOTE: Make sure to execute above command on the node image as well.

Log out and log back in to get a shell environment with Lmod now having system-wide access.
You can check this with :
`$ module avail`

Task 3: Install Spack (5 min)

Assuming you have already installed the dependencies for *spack* as per Task 1, download the source code from the official repository and extract the tarball.

```
$ wget https://github.com/spack/spack/releases/download/v1.2.2/spack-1.2.2.tar.gz
$ tar -xf spack-1.2.2.tar.gz
```

Copy the default configuration file (`config.yaml`), so it can be modified freely.

```
$ cd spack-1.2.2/etc/spack/
$ cp defaults/base/config.yaml .
```

Modify the entry `install_tree:root` to `/nfs/sw/spack`

Next, create a new file with name `modules.yaml` in this very folder with the following content

```
modules:
  default:
    enable:
      - lmod
    roots:
      lmod: /nfs/sw/modules
    lmod:
      hash_length: 0
      hierarchy:
        - mpi
      core_compilers:
        - gcc@11.5.0
```

spack is run from source, i.e no compilation is necessary. However, to setup the spack environment for a terminal session, one needs to do:

```
$ source ./spack-1.2.2/share/spack/setup-env.sh
```

To make sure *Lmod* finds the packages from *spack*:

```
$ echo '/nfs/sw/modules/linux-rocky9-x86_64/Core' > /nfs/sw/lmod/lmod/init/.modulespath
```

Task 4: Install your first package (5 min)

In this task, attempt to install the archiving program **GNU Tar**.

- look at the detailed info of that package by: `$ spack info tar`

- Inspect the concretized specification with its dependency tree by : `$ spack spec tar`
- Install the package by : `$ spack install tar`

Task 5: The module command (5 min)

After Task 4, our system has two different versions of the package `tar`. In this task, verify if you can dynamically switch between the two. If the configuration of Lmod and Spack is correct, `spack install` has created module files that we can use with Lmod. Try to understand the output from the following order of commands.

```
$ tar --version
$ which tar


$ module avail tar
$ module load tar
$ tar --version
$ which tar


$ module unload tar
$ tar --version
$ which tar


$ module purge
```

Task 6: Install a software suite (5 min)

In this task, attempt to install **Gromacs**, an open-source high performance molecular dynamics code.
NOTE: It could take some time to compile the actual software and its dependencies.

Inspect the dependency tree with: `$ spack spec -I gromacs`
Install the package with: `$ spack install gromacs`

While the package is being installed, open a new terminal window, log into your cluster and continue with the remaining tasks.

Task 7: Spack specifications (10 min)

Try to interpret the following specifications
1. `tar zip=gzip ^bzip2+debug ^xz+pic`
2. `gromacs +blas +cuda`
3. `gromacs%gcc@12.2.0 ^openmpi%gcc@11.5.0`

Use `spack spec -I` to check if you could install the above specs, and which previously installed packages could be reused.