SFT Spack Plans

Graeme Stewart and Javier Cervantes Villanueva

2018-11-07

Future Plans for Stack Building

- SFT group would like to investigate other approaches to building software stacks
- Very strong alignment with this HSF group
- Spack continues to be one of our most promising candidates
    - Clearly satisfies a lot of the use cases we identified ‘out of the box’
    - But the new features that are arriving are making the box larger and more comfortable…
        - Spack chains we discussed a lot
        - Todd has also indicated a more fuzzy concretization strategy will come
            - Avoid rebuilds that are not really needed because a ‘good enough’ package build exists
    - Synergy with rest of HEP and scientific software community is very attractive

LCG Spack Prototype

- Goal is to build fairly complete version of the current LCG release with Spack
- But the real goal is to investigate that as a base for building other software
  - Experiment production software
  - Developer build environment
- It is here that the use cases get more interesting
- Starter platform
  - CentOS7 using gcc8.2.0 compiler
- Package sets for…
  - All Gaudi externals (HEP packages: ROOT, CLHEP, HepPDT, HepMC, AIDA, RELAX)
  - FCC software (which can then exercise Spack chain in particular)
  - ATLAS externals (with current ATLAS CMake recipe)

Bootstrap...

- Use a base C7 bootstrap image, graemeastewart/c7-dev
- Use the a recent version of Spack develop branch
  - In principle HEAD, should probably update regularly
  - Apply Spack chain PR (#8772)
  - Apply much improved ROOT recipe (#8428)
  - https://github.com/graeme-a-stewart/spack/tree/hsf-spack
- I started all of this using docker on my OS X laptop
  - Mainly as I am an obsessively local developer!
  - Mounting Spack itself from the main laptop filesystem
  - Use a different mount point for the build and install areas
- Developed configuration we will put into this repo (nothing there yet)
  - https://github.com/HSF/spack-config
  - And might be interesting to share more widely later

A Cautionary Tale

- First attempts came to naught in a most frustrating way…

```
"terminfo.tmp", line 1358, terminal 'nsterm': alias Apple_Terminal multiply defined.
E/Eterm: No such file or directory
"terminfo.tmp", line 5924, terminal 'Eterm': can't open
/opt/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/ncurses-6.1-rqw3cmcwvd76h4rjgtaxpvrqg47prwkg/share/terminfo/
E/Eterm
? tic could not build
/opt/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/ncurses-6.1-rqw3cmcwvd76h4rjgtaxpvrqg47prwkg/share/terminfo
make[1]: *** [install.data] Error 1
make[1]: Leaving directory `/tmp/root/spack-stage/spack-stage-JHpBeY/ncurses-6.1/build_ncursesw/misc'
```

- Eventually, after learning a lot about termcap’s install process I realised the problem was the underlying filesystem
  - Apple’s APFS filesystem defaults to case-insensitive, case preserving
    - This messes up the ncurses build
- Solution: use a disk image that is explicitly case sensitive
  - Yes, this is annoying and definitely less convenient

After that...

- Bootstrap to build gcc8.2.0 from the system compiler (gcc4.8.5) works fine
- ROOT build works fine
  - Minor silly issues with originally missing C++ system compiler and funny tar configuration option (`export FORCE_UNSAFE_CONFIGURE=1` as a work around)
- Almost all other Gaudi HEP packages all in Spack already and build fine
- RELAX is missing...
  - Adding this right now, should be easy as it's a very straightforward CMake package
- This should complete Step 1 - Gaudi dependencies installed

- So, just starting up this work so ***watch this space***