

Packaging at CHEP

Graeme A Stewart

2019-12-11

Tools and Packaging at CHEP

- Part of Track 5, Software Development
 - <https://indico.cern.ch/event/773049/sessions/323858/#20191105>
- Four talk slots shows that packaging is still considered an interesting problem for HEP
- Caveat Emptor - what follows is my own view!

Modern Software Stack Building for HEP	Graeme A Stewart 
<i>Riverbank R2, Adelaide Convention Centre</i>	14:00 - 14:15
Gentoo Prefix as a physics software manager	Prof. Benda Xu 
<i>Riverbank R2, Adelaide Convention Centre</i>	14:15 - 14:30
SpackDev: Parallel Package Development with Spack	Elizabeth Sexton-Kennedy 
<i>Riverbank R2, Adelaide Convention Centre</i>	14:30 - 14:45
Sustainable software packaging for end users with conda	Chris Burr 
<i>Riverbank R2, Adelaide Convention Centre</i>	14:45 - 15:00

Spack and SpackDev

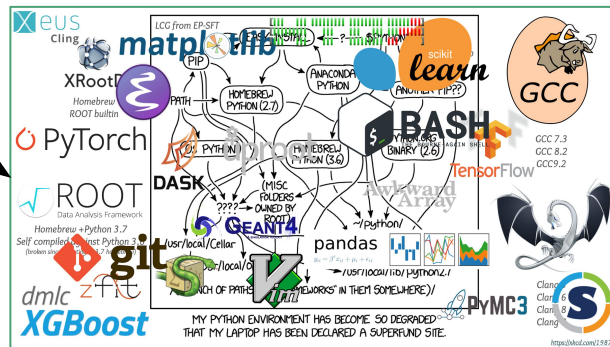
- My talk concentrated on work we already presented in the packaging WG
 - Update on the group's activities since Ben's presentation in Sofia
 - That was appreciated, especially reviewing why the problem is more complex than it naively appears
 - I reported on the active Spack work in SuperNEMO and FCC and the SFT summer project
 - Thanks to Ben, Javier and Hobbs!
 - Part played in the turnkey software stack was useful to discuss
 - Big bonus in the discussion was to hear that FAIR experiments have adopted Spack
- SpackDev talk (from Chris++, given by Liz)
 - Commitment to Spack in the ECP (Exascale Computing Project) was underlined
 - Reinforces benefits of working with a larger scientific community
 - SpackDev as an idea, to setup a multi-package development area, had to be explained quite carefully
 - Repository independence of neutrino software packages in LArSoft
 - ATLAS and CMS have a 'fat' package for Athena and CMSSW, respectively
 - Could be interesting for LHCb

Gentoo Prefix

- Presented by Benda Xu
 - Guilherme (friend of our working group) was co-author
- Also well known in the packing group as a potential solution to the packaging problem
 - Advantages of very deep build (to libc and ld) are to gain complete independence from the underlying distribution
 - Almost like a container without containers!
 - Mature system, lots of recipes
 - Independence of each prefix...
 - No sharing of lower level components of the build
 - Hard to patch releases (update a library, add a new package) without reinstalling everything
- IMO: Powerful tool, but not quite the best fit for the full suite of our use cases

Conda

Chris' visual summary
of the problem!



- Conda also well known to us in WG
 - Presentations at JLab from Anaconda team, also from Chris to this group
- What I like about this is that it's very widely used by *users*
 - As opposed to librarians, package managers and enthusiasts
- Ergonomics are excellent
- Fit would be for delivering flexible and up to date (bleeding edges!) to **analysis users**
 - Don't have to carry the weight of the full production stack around
 - Nor burden the production stack with latest version of PyTorch
- Questions of capturing this for analysis development and preservation would need to be looked at
 - Pieces seem to be there, would be good to try out...

Final word from the Track 5 summary from Martin...

Spack Package Manager



Picking up momentum in HEP

- used by FCC, Key4HEP, SuperNEMO
- FAIR moving to spack
- CMS has proof of concept
- ATLAS considering but things still to be understood
- LHCb and Belle II willing to follow



spack.io

SpackDev: Multi-Package Development with Spack

“Coordinated build & test for integration, or initialize an environment for rapid build & test cycles of a particular package.”

- extension to Spack to help with development of interdependent packages