

Spack Status Report

Graeme Stewart and Marco Clemencic

Reminder of the Idea

- We discussed back in the AF of June if there was a possibility that, as a community of NHEP experiments, we could identify and share a common build tool
 - Specifically was [Spack](#) a good fit for that role?
- We knew that there had been investigations into Spack from different parts of the NHEP community (more than any other tool we knew of)
 - So we wanted to understand
 - How much progress had been made?
 - What problems have come to light?
 - Are these being worked around? Or fixed?
 - Is Spack in production; or planned to go into production?

Survey

- We didn't do an exhaustive survey, but we did speak to quite a few communities at CERN and outside
 - FAIR
 - Fermilab
 - HSF Packaging Group
 - SFT
 - ATLAS (Attila)
 - ALICE (Giulio)
 - LHCb (Marco)
- CMS had already reported on their experience with Spack (they didn't like it)

Round up of the most important points I

Communities open to collaboration

- FAIR
 - Successfully using Spack to build software stacks for their HPC containers
 - Spack built data products go to CVMFS, but then get pointed to by the container
 - Have not successfully migrate to evolve stack building for experiments to Spack
 - CBM came close, but had issues with their own software and did not complete the migration
 - Build times with Spack were longer than with the old tool (also seen by CMS), because of building “system” packages
- Fermilab
 - Have not managed to migrate LArSoft to Spack
 - Seem to be rather stuck at the development story, which is not Spack’s strong point
- HSF Packaging Group
 - Report on packaging issues still very relevant (including use cases)
 - It is probably safe to assume that there is no one size fits all, so different experiments have different priorities

Round up of the most important points II

- SFT
 - Successful deployment of Spack for Key4hep stack
 - A number of difficulties worked around
 - Still using LCGCmake for LCG releases
 - Significant interest in switching to Spack for LCG releases
 - **RPMs would be needed from a Spack build to harmonise with the current system**
- ATLAS
 - No plans to switch to Spack at the moment
 - Current way of ingesting LCG releases, RPMs, would need to stay, but could be *simplified?*
- ALICE
 - Very happy with AliBuild, no plans to change
 - See significant advantages over Spack, in particular aware of long term “costs”
- LHCb
 - Serious interest in using a tool like Spack for the LHCb builds
 - But would not go alone - *SFT support would be essential*

Observations

- Spack is still under development and there have been many useful improvements
 - New concretiser
 - Improved binary caches
 - Spack env
- Deploying a production instance with Spack means freezing at some point
 - N.B. this both freezes the software and the recipe set (albeit patching can be done, e.g., an updated recipe)
 - This helps with sensitivity to package hashes
- Specifics of recipes are controversial
 - Single recipe grows cumbersome (deprecate old releases/options?)
- Relocation issues can be worked around, but why do they still exist?
 - We don't really understand completely why this is a problem (Spack issues? nasty packages?)
 - Could almost certainly be fixed (cf. Alibuild)
- Spack is *good at deploying production stable software (on clusters)*
 - Not so good at the “developer story”, but is this your critical use case?

Tentative Conclusions

- Spack is not a panacea
- It does do some things well
 - Build and deployment of stable production stacks
 - Large community contributing to recipes, to which we contribute NHEP specifics
- There are some things that it's not so good at
 - Relocation seems slow (binary string search and replace) and a bit buggy right now
 - Development process doesn't seem very well supported
 - Overheads to the build time are non-negligible
- And some things just divide people
 - Monorepo mixing recipes and software
 - Single recipe per package
 - Use of system packages
- So can we agree on a single build tool?
 - Highly unlikely, given the reality of existing solutions and cost/benefit of migration
- Perhaps the real question is then is Spack better than what we have today for a *critical mass of NHEP projects*
 - A community of multiple Spack users can then help to lower barriers and costs, which we, to some extent, have