

Optimizing experiments' software stack management with Spack

CERN Summer Student 2018 - Paul Chelarescu



Javier Cervantes Villanueva
EP-SFT

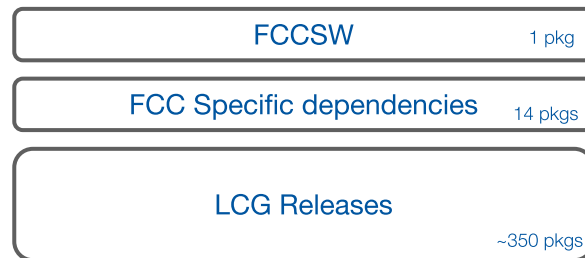


Introduction

FCC software is built against LCG's CVMFS installation

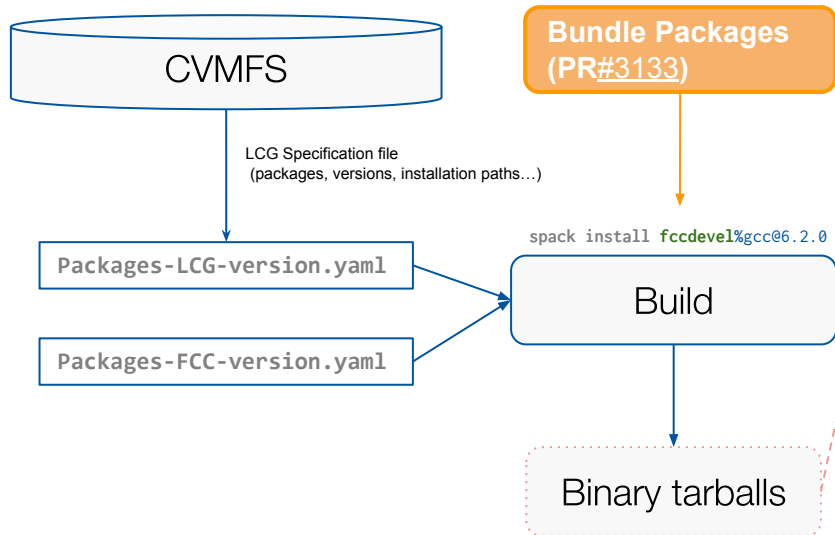
- First step: create **packages.yaml** file describing LCG specs
 - Using LCG compiler and packages description files
- Allows to build with Spack against LCG stack
- Specific package versions might replace LCG ones
- Package definitions on github:
 - Separate HEP-FCC/**fcc-spack** for fcc-specific packages
 - Using HEP-SF/**hep-spack** as a basis
 - Spack **builtin**

Emulating
incremental
build

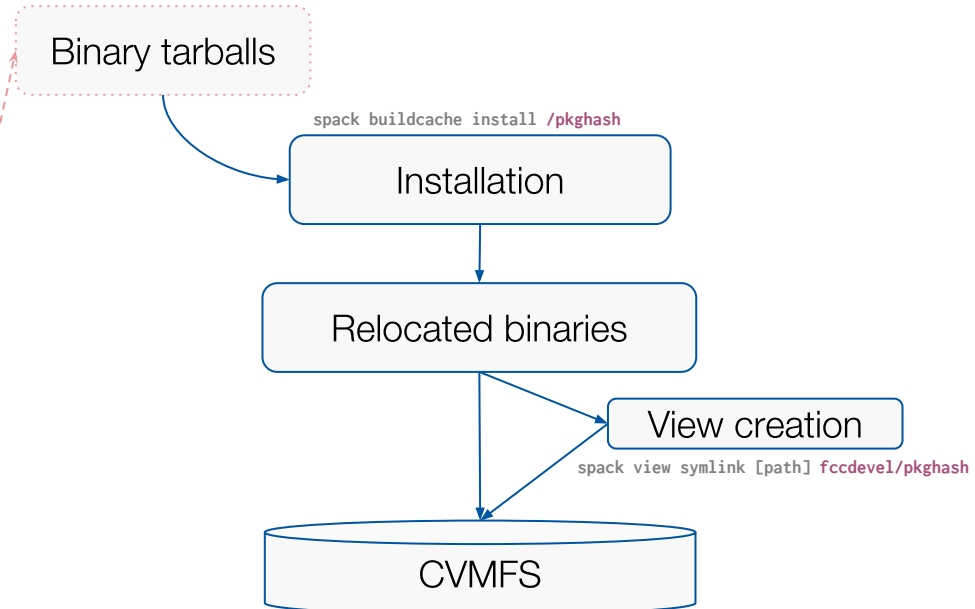


Workflow

Build node



CVMFS Stratum 0 Node



Next goal: Speed up builds

Reduce redundant work repeated every day

Last report
07/02/18

Try to not build what is already installed in CVMFS, incremental approach

- Options
 - Custom `packages.yaml` file (LCG Releases approach)
 - ▼ Manually scan through different CVMFS paths (not scalable)
 - ▼ Does not consider different hashes
 - Binary repository with all possible binaries/combinations installed so far
 - ▼ Still requires download and installation of binaries
 - Read from remote `opt/spack/.spack-db/index.json`
 - ▼ Not writable from CVMFS
 - ▼ Not easy to synchronize
- Desirable
 - Given an external path, automatically find out and consider matching hashes
 - Discovered packages might be linked as they were specified in the `packages.yaml`

Updates

Spack ccache

Merged

[PR #3761](#)

Status

- Development started by @junghans - 7 April 2017
- Merged - 9 July 2018

```
config:  
  ccache: true
```

Features

- Speed up recompilations
- Easy setup: one single line in config.yaml
- Good integration in Jenkins
 - Last used node distribution algorithm
- Good performance
 - ~30% time reduction for the FCC Software stack

[Credits: Summer Student Report - Paul Chelarescu](#)

Spack chain

WIP

[PR #8772](#)

Status

- Development started by @amundson [#8014](#) - 3 May 2018
- Multiple contributors:
 - [#8014](#) (amundson) → [#8545](#) (gartung) → [#8772](#) (scheibel, paul-chelarescu)
- Still opened - Last edition 11 days (scheibel)

Features

- Chain of spack install trees
- Symlink to packages existing in different installations
- **Incremental builds** using local/remote Spack trees
 - Functionality already provided by LCGCMake
 - Key feature to maintain large sets of packages in an efficient way

Spack chain

WIP

[PR #8772](#)

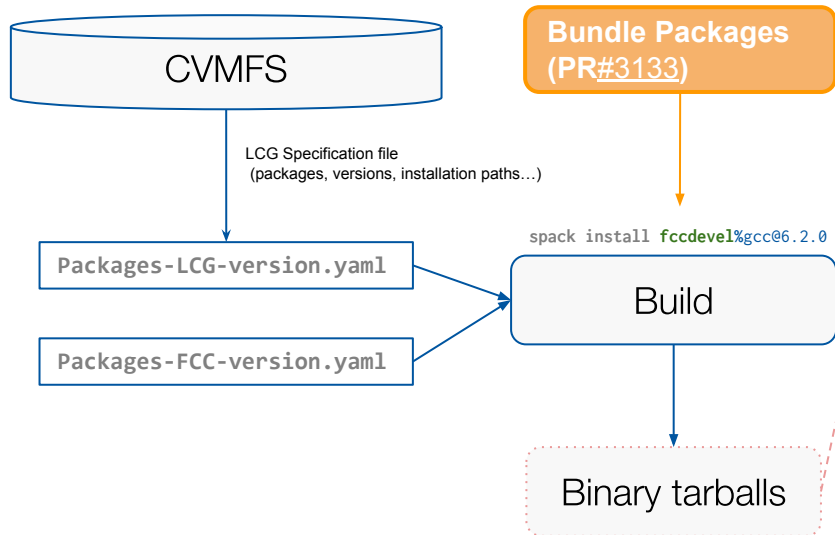
Use cases [1]

- A two-link chain:
 - Packages are provided by a system-level spack installation.
 - The user adds packages to his/her own private spack installation.
- A three-link chain:
 - Same as above, but with a group-level installation in between the user and system.
- FCC case (*a two-link chain*):
 - CVMFS remote spack installation + build node installation
 - Reduce waste of time building same package, same version

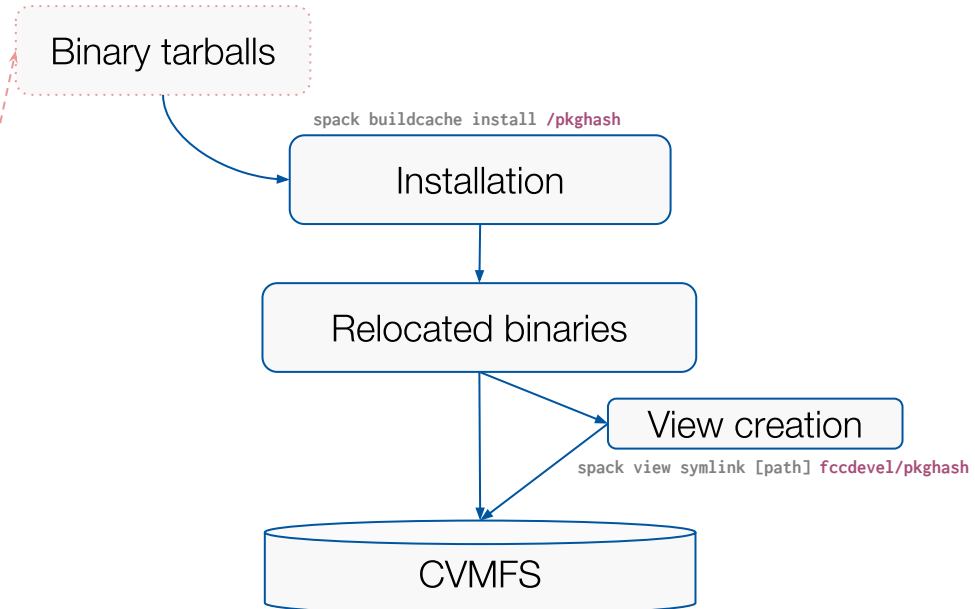
[\[1\] James Amundson #PR 8014](#)

Workflow

Build node



CVMFS Stratum 0 Node



FCC Preliminary results

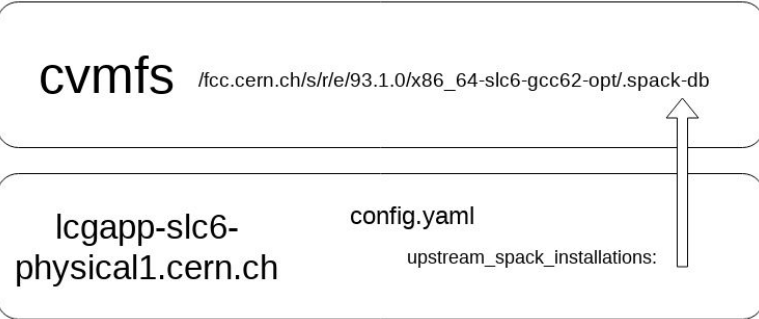
```
$ spack install fccdevel
==> acts-core is installed in an upstream Spack instance
==> podio is installed in an upstream Spack instance
==> fcc-edm is installed in an upstream Spack instance
==> fcc-physics is installed in an upstream Spack instance
==> gaudi is installed in an upstream Spack instance
...
==> git is installed in an upstream Spack instance
==> Installing py-setuptools
Fetch: 0.57s. Build: 1.97s. Total: 2.54s.
==> heppy is installed in an upstream Spack instance
==> Installing papas
Fetch: 2.52s. Build: 43.22s. Total: 45.74s.
==> tricktrack is installed in an upstream Spack instance
==> Installing fccdevel
Fetch: 1.76s. Build: 2.02s. Total: 3.79s.
```

- Only install what is needed
- Significant reduction in build time
 - **1 minute vs 52 minutes**
- When is fccdevel in cvmfs
 - 28 seconds

cvmfs /fcc.cern.ch/s/r/e/93.1.0/x86_64-slc6-gcc62-opt/spack-db

lcgapp-slc6-
physical1.cern.ch

config.yaml
upstream_spack_installations:



[Credits: Summer Student Report - Paul Chelarescu](#)

CDash integration

Non-working

- Feature present and working at some point
- Currently spread on different opened PR's
 - [PR #7114](#)

Conclusions

- FCC software infrastructure is currently built and deployed using Spack
 - Spack fully covers our basic requirements
 - Additional scripts needed to complete the workflow
- Recent Spack developments
 - **Speed up** our builds as well as allow us to be **more efficient** (resources / time)
 - Make Spack a more robust Package manager
 - Long-awaited features for many users
- Great progress as a result of **Paul's contribution** during this summer