

Librarian and Integrators Workshop: Summary

Patricia Mendez Lorenzo
EP-SFT Group Meeting
CERN - 2 July 2018

Intro

- On the 30th of May we met the experiments/users with the project evolution as the main goal in mind
- But first, let me introduce our users:
 - YOU ALL
 - ATLAS, LHCb, BE, FCC, Hadoop, SWAN, Spark
 - Lots of individual users

Some highlights of the Workshop

- Quite a lot of people
 - All experiments/users represented
- Very dense agenda
 - Too much at the end...
- Long discussion session included in agenda
 - Looooong day

Workshop agenda

| | | | | |
|-------|---|------------------------------------|---------------|--|
| 09:00 | Welcome and workshop expectations 32-1-A24, CERN | <i>Pere Mato Vila</i> | 09:00 - 09:10 | |
| | What SFT is providing 32-1-A24, CERN | <i>Patricia Mendez Lorenzo</i> | 09:10 - 09:40 | |
| | Build nodes configuration: The role of HEP_OSlibs 32-1-A24, CERN | <i>Andrea Valassi</i> | 09:40 - 09:55 | |
| 10:00 | Releases and nightlies validation: LCGtest 32-1-A24, CERN | <i>Rafal Pacholek</i> | 09:55 - 10:10 | |
| | Generators models; The Genser contribution 32-1-A24, CERN | <i>Dmitri Konstantinov</i> | 10:10 - 10:40 | |
| | Coffee Break 32-1-A24, CERN | | 10:40 - 11:00 | |
| 11:00 | The SWAN project: current model and future requirements 32-1-A24, CERN | <i>Diogo Castro</i> | 11:00 - 11:30 | |
| | Hadoop and Spark Service: Current usage of CVMFS and future requirements 32-1-A24, CERN | <i>Prasanth Kothuri</i> | 11:30 - 11:40 | |
| | The FCC project: current model and future requirements 32-1-A24, CERN | <i>Javier Cervantes Villanueva</i> | 11:40 - 12:10 | |
| 12:00 | Lunch Break | | | |
| 13:00 | CERN | | 12:10 - 13:10 | |
| | | | | |
| | | | | |
| | | | | |
| 14:00 | The ATLAS experiment: current model and future requirements 32-1-A24, CERN | <i>Emil Obreshkov</i> | 14:05 - 14:35 | |
| | The LHCb experiment: current model and future requirements 32-1-A24, CERN | <i>Ben Couturier et al.</i> | 14:35 - 15:05 | |
| 15:00 | The BE community: current model and future requirements 32-1-A24, CERN | <i>Juan David Gonzalez Cobas</i> | 15:05 - 15:35 | |
| | The ALICE experiment: Future requirements 32-1-A24, CERN | <i>Giulio Eulisse</i> | 15:35 - 15:50 | |
| 16:00 | Coffee Break 32-1-A24, CERN | | 15:50 - 16:10 | |
| | The CMS experiment: Future requirements 32-1-A24, CERN | <i>Shahzad Malik Muzaffar</i> | 16:10 - 16:25 | |
| | Proposal for a LCG based analysis ecosystem 32-1-A24, CERN | <i>Danilo Piparo</i> | 16:25 - 16:35 | |
| | AFS Phaseout status 32-1-A24, CERN | <i>Jan Iven</i> | 16:35 - 16:45 | |
| 17:00 | HSF Packaging Group: Common Directions for the Future 32-1-A24, CERN | <i>Graeme Stewart</i> | 16:45 - 17:15 | |
| | Final discussion, conclusions and agreements | | | |
| 18:00 | CERN | | 17:15 - 18:15 | |

Intro (I)

Goals for this Workshop

- ✦ An opportunity to re-orientate the SFT activities to better server the internal and external projects
- ✦ Review the status of the SPI project
 - ✦ Compare what is provided versus what is currently needed or will be needed in the next 2-3 years (for Run3 after LS2)
 - ✦ Identify the strong points and weaknesses
- ✦ Listen to the LHC users and other communities to identify common areas of interest
 - ✦ We have tried to cover in the agenda all LHC experiments and current clients
- ✦ Establish a common strategy that will cope with the new requirements of the LHC users and other communities
 - ✦ Aligned with HSF common directions if possible

All the ingredients to define the SPI program of work during LS2

Welcome talk
given by Pere

Intro (II)

(Some) Topics to discuss at the end of the day

- **AFS common deprecation strategy**
 - Where are we at this moment?
- **Evolution of build nodes**
 - Role of HepOSlibs and its evolution
- **Releases/software distribution**
 - evolution and current status
- **arm64 strategy**
- **Future of LIM**
 - How would you like to focus it
- **Anything you want to bring to the table**

My talk
given just after

HEPOS_libs

- IT IS AN IMPORTANT TOPIC not sufficiently treated yet

What is HEP_OSlibs?

a meta-package: *'yum install HEP_OSlibs'* on RedHat is equivalent to
'yum install alsa-lib-devel.x86_64 ... zsh.x86_64'

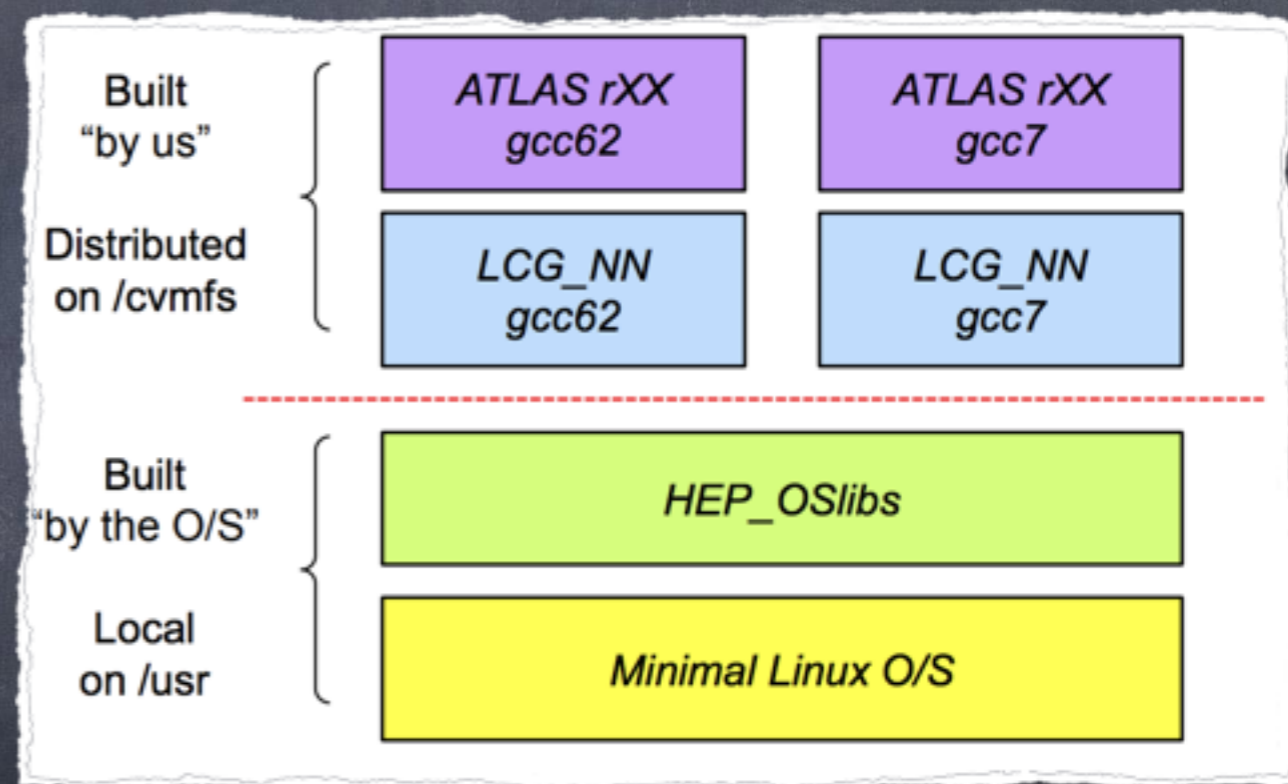
- a list of packages
- without a specific version number
- taken from the official O/S repos (not from EPEL)
- "the O/S level dependencies of the LHC experiment software stacks"
- *a more precise definition of what a "supported O/S" is*

Andrea
Valassi

- It should be triggering more headaches of what it is doing!
- Available in Grid nodes, build nodes, SWAN, lxplus
- It affects us directly:
 - Our build nodes and results
 - Therefore our reproducibility and exportation capability
 - ... personally, my quality of life

HEPOS_libs

- Basically the problem is that this boundary is not yet defined
- A not clear boundary can drive to conflicts at run/build time
- Available for slc6/Centos7/Ubuntu



Andrea
Valassi

HEPOS_libs

- Andrea Valassi' statement in short: "If you can take a package from O/S why rebuild it yourself?"

On Linux, I see only 3 reasons to build a package ourselves (LCG stack):

- 1. Need a different C++ version than the O/S compiler (e.g. Boost)
- 2. Need a newer package version than the O/S version (e.g. Python)
- 3. Need a HEP or EPEL package not available in the O/S (e.g. ROOT)
- In all other cases, I would install missing O/S packages via HEP_OSlibs

Andrea
Valassi

- My opinion: HE IS RIGHT
 - If you have to build many packages, believe me; you want HEPOS_libs → Qt5 experience
 - And you do not want a tiny HEP_OSlibs just by principle
 - You do not want to build X11 like packages

HOME-MADE PARTS OF SPI; LCGTESTS

- **Rafal Pacholek' BABY**

- New project available in gitlab aimed to validate the SW we provide after distribution to CVMFS

Why do we need it?

- Test if required functionality is working in custom installations
- Test if numerical software gives the same result
- Test if new versions of packages break other packages

Rafal
Pacholek

- **HOWEVER:** Without the experiments/users we are blind
- The repository needs to be opened to them and they should contribute with their own tests

LCGTESTS

- Uses CTest as framework and already is sending results to CDASH
- Current tests delivered to Python packages

However... based on the developer point of view, we cannot escalate:

- Setup of all scripts in cmake from scratch
- No integration with other tests FW
- No modularity
- Things are black or white only...

Still there is hope...

- Rafal proposed AVOCADO
 - Developed and used by RedHat
 - able to fulfil all deficiencies that the current service has

HOME-MADE PARTS OF SPI: Genser

- Presented by Dmitri Konstantinov

- Much much more than generators providers

- Views code creators

- Binaries reallocation software managers

- LCGENV creators:

- Python sw which enables the correct runtime environment

- It reads in release metafiles and spits out environment for given package

- Implementation of additional non-standard environment variables is possible

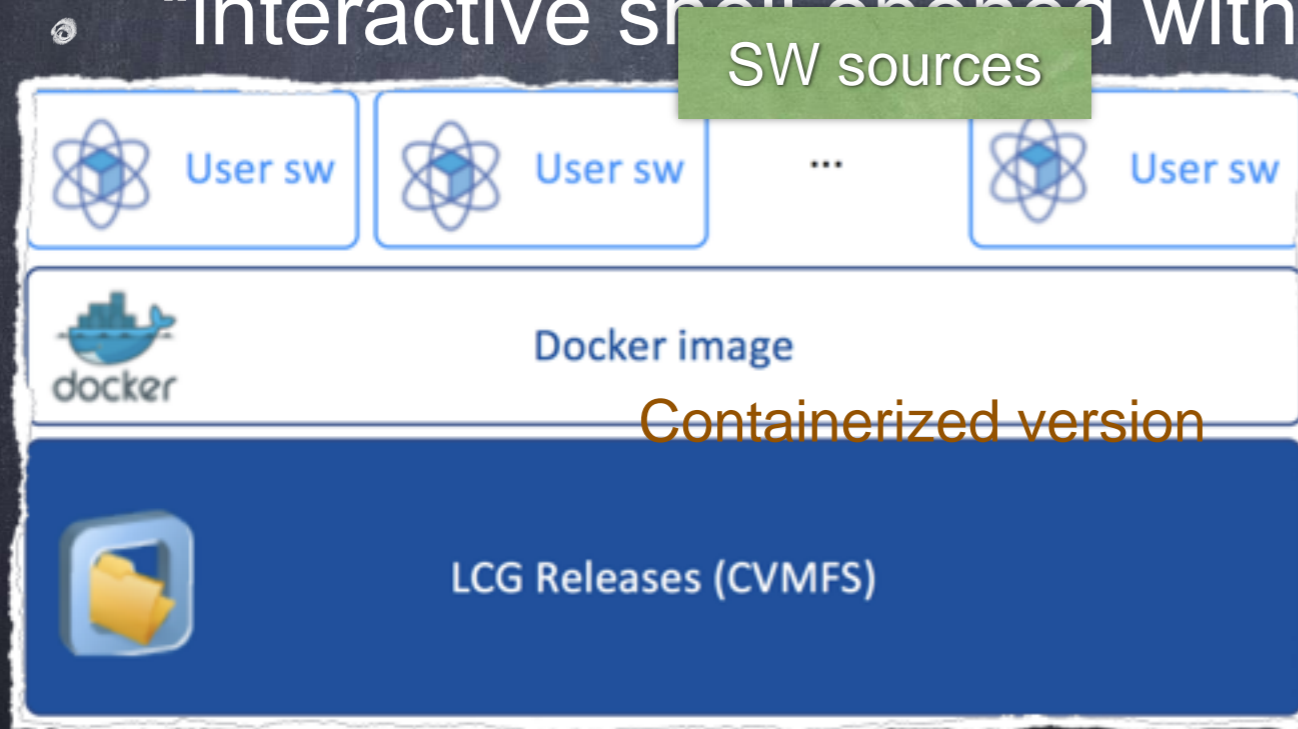


Key team of
SPI

Our users: SWAN

Presented by Diogo Castro.

“interactive shell opened within the browser”



Stability is their major request

- > SWAN bet on LCG Releases from day 0
 - Removed the need for installation and configuration of packages
 - Reduced the Docker Images size
 - Provides Jupyter kernels
- > Multiple stacks
 - Python 2 or Python 3
 - Bleeding edge

Not only they bet on LCG Releases:
SWAN uses the nightlies in all their extension

Our users: Hadoop

- Presented by Prasanth Kothuri
 - IT Service running and setting infrastructure for (primarily) Apache FW and Big Data Ecosystem
 - Starting the SWAN collaboration in 2016:
 - SWAN session (container) to act as Hadoop and Spark client
 - Two simple steps then:
 - Distribution of Hadoop and Spark binaries through LCG releases
 - Distribution of the cluster specific configuration from CVMFS
 - Requirements:
 - New PKGs versions, multi-versions support

Prasanth
Kothuri

• Apache Spark is actively being developed and new versions are released at **short timescales**, which most of our users would like to use, so a **stable working bleeding edge** would be very nice

“latest” view is ideal in this case

Our users: Hadoop

Access to Hadoop and Spark service from lxplus

- CERN user with authorization to Hadoop and Spark service can run spark jobs and hdfs commands thanks to the software in LCG Releases and the configuration in CVMFS
- KB: <https://cern.service-now.com/service-portal/article.do?n=KB0004426>

Users of PySpark (python on spark) profit from the huge number of packages available in python distribution. The service offloads maintenance of python distribution to LCG Releases

Hadoop and Spark service relies on LCG Releases to deliver stable Hadoop-client and configuration to our user community

LCG Releases are made available on the service nodes by 1) Installing HEP_Oslibs and 2) mounting sft.cern.ch using CVMFS

Two simple orders:

- LCG release (view) setup
- Cluster configuration (cvmfs) setup

Prasanth
Kothuri

Our users: FCC

Presented by Javier Cervantes

Quite in line with LCG Releases

LCG Releases and FCC

Motivation

- LHC common software
 - We adopt the same set of tools, standards and procedures as the rest of LHC experiments
 - New developments can be integrated from/by existing projects
- Evolving our software stack continuously
 - Based on *user* needs and current state of the art
 - New packages constantly added
 - Version maintenance
- Reduce complexity
- Speed up FCC-specific builds

Works in general quite well

Javier
Cervantes

FCC use of LCG Releases

Summary of the LCG Services in use

- ✓ Releases
 - FCC Release on top
- ✓ Nightlies
 - FCC Nightlies on top
- ✓ Views
 - FCC Views source it in first place
 - Setup FCC Environment in SWAN
- ✓ lcginfo.cern.ch
 - Check content releases/nightlies
- ✓ LCG contrib compilers
 - Spack / FCC views take compilers from lcg/contrib in cvmfs
- ✗ RPM's / Tarfiles
 - LCG Packages directly taken from CVMFS
 - Spack requires its own binary format (due to metainformation)
- ✗ Basic Docker images
 - Not yet, possibly in a future
- ✗ Docker containers (with LCG Releases)

Javier
Cervantes

- Performance and efficiency are the most important requirements
- Improvement proposals for the views

Our users: BE

- Presented by Juan David Gonzalez
- Latest community entering the project
 - Interested in Python3 builds with about 30 new packages/versions requested (added in LCG_93)
 - Very conservative in releases deployment (LCG_93 for 2018)
 - Challenge: No AFS, no CVMFS, no EOS in TN
 - Relocation for them is really THE POINT
 - LCG_93 feedback: Fully operative and implemented in their SW infrastructure

Our users: BE

- The current model seems to be satisfactory and LCG more than enough for current needs.
- It *might* be that requirements for use of Geant4 or ROOT appear (unlikely).
- EOS is slowly getting to the TN; this might change things somehow.
- Integration with CI infrastructure of LCG *must* happen
- Integration with CI infrastructure of BE-CO *must* be studied/defined
- Lots of interest in ARM architecture support (Zynq serious candidate for the Next Platforms Contest)

Juan David
Gonzalez

As everybody else...
let me explain you the current
situation

LET'S GO TO LHCB

- Presented by Ben Couturier
- LHCb build system is based in [Gaudi](#) (cmake build with a toolchain.cmake file to manage dependencies)
- Quite recognisable setup to what SFT has
- Indeed, [the help provided by Ben and Marco to our system has been unplayable in many occasions](#)
 - Adoption of LHCb sw solutions for RPMs handling, etc
 - Contributions through merge requests have been always welcome

LHCb setup

- Externals taken from the LCG stack
 - specific LCG_XX.txt used to identify dependencies
- LHCb releases: custom RPMs created in Jenkins, stored in EOS
 - LCG externals dependencies based on the set of RPMs we provide
 - Distribution/installation in CVMFS using the SFT yum repository
 - post-install relocation

LHCB

Platforms/compilers

- SLC5/SLC6/Centos7 and ubuntu to small scale
- gcc62 (2018 stack), gcc7, gcc8 (Run3)
- clang is also requested by the experiment
- HEPOS_libs needed on top of base system for SLC/Centos

LHCB continuous integration system

- Quite similar to what SFT also has
 - Jenkins orchestration structure
 - Openstack VMs
 - Docker containers
- Builds based on LCG releases + dev3 + dev4 (LCH nightly slots) —> results copied to CVMFS

Feedback and Issues

Ben Couturier

Use within LHCb

- LCG stack's very controlled aspect works well for current production software (reconstruction, simulations, trigger)
 - Generators are more difficult to deal with as they evolve separately
- LHCb Production releases are not dynamic/flexible enough for Analysis
 - installing latest version of Python modules on top is not very easy
 - Some analysts have their own stacks (conda...)
- Our Grid middleware does not easily fit in the framework and has to be prepared separately
 - We even have incompatibilities between the dependencies (e.g. Boost for GFAL2)
- Would be nice to have finer granularity in some packages e.g. for ROOT, c.f. Debian release example
- Running/Rebuilding old stacks can be problematic (e.g. `rebuild`)

e.g. Run of the old version of the trigger:
incompatibilities between the xrootd LCG version and the WLCG server

Issues

- Separate build system for LCG and LHCb complicates matter e.g. port to ARM or PowerPC
- Base platform definition unclear: Need HepOS lib RPM
 - N.B. Even that is not an exhaustive list of packages
- Not easy to distribute the work of integrating new externals
- Difference in release cycles between externals and generators complicates the management of the stack

Some problems but still a lot easier and smoother than in the CMT, pre-RPM days

LHCB requirements

- One tool for externals + LHCb SW (ideally)
 - HSF direction
- Ease the build of other externals on top of LCG
- BETTER DEFINITION OF BASE SYSTEM
- Management of multiple install areas
 - Packages removal with their dependencies
 - Minimal installations for specific purposes

LET'S GO TO ATLAS

Presented by Emil Obreshkov

LCG releases in use (1/2)

- Validation of MC software (Sherpa2.2.5, Herwig7.1.3) in some old releases (SVN & CMT).

| <u>Release series</u> | <u>LCG configuration</u> | <u>SVN/GIT</u> | <u>compiler</u> | <u>purpose</u> |
|-----------------------|--------------------------|----------------|-----------------|-------------------------|
| 19.2.X.Y | lgcmt_67c | SVN | gcc47 | MC15 production |
| 20.7.9.9.Z,MCProd | LCG_81f | SVN | gcc49 | validation |
| 21.0 | LCG_88 | GIT | gcc62 | being prepared for MC16 |

- 67c and 81f LCG releases still used from AFS during our nightly builds, aim is to move away from that at the end of the year, depending on validation success and issues.

Move MC production to 21.X (using LCG_88 from cvmfs).

Emil
Obreshkov

ATLAS setup

- Quite in line with the SFT
 - High dependency in RPMs in releases —> high stability needed
 - Nightlies based on our dev3
 - However new ROOT validations are based on dev4
- Requests
 - RPMs for nightly builds —> Otherwise no validation of software
 - pre-releases
 - Jira active use

LET'S GO TO POTENTIAL USERS: CMS

Presented by Shahzad Muzaffar

- 8 Open release cycles
- 360+ externals
- Using SCRAM (perl + Make) to configure/build CMSSW
 - CMAKE evaluated again, however it has a too high overhead
 - OK for releases, but not scalable for user dev on shared FS
 - SCRAM was re-written in 2008 to reduce the overhead
- Package based in RPMS using their own repository → CVMFS distribution

CMS Model

- Software stack available for:
 - SLC6/Centos7
 - AMD64/AARCH64
 - GCC6/7/8, ICC, LLVM
- CMSSW integration builds with externals executed twice per day
 - Including all cycles/architecture
 - Full build with each new external or it's a new week
 - Build of patch release only if CMSSW code changed

Possible interactions

- CMS is betting for ML
 - SFT provides a full ecosystem of ML tools including Tensorflow
 - This could be a common point to share build structures

LET'S GO TO POTENTIAL USERS: ALICE

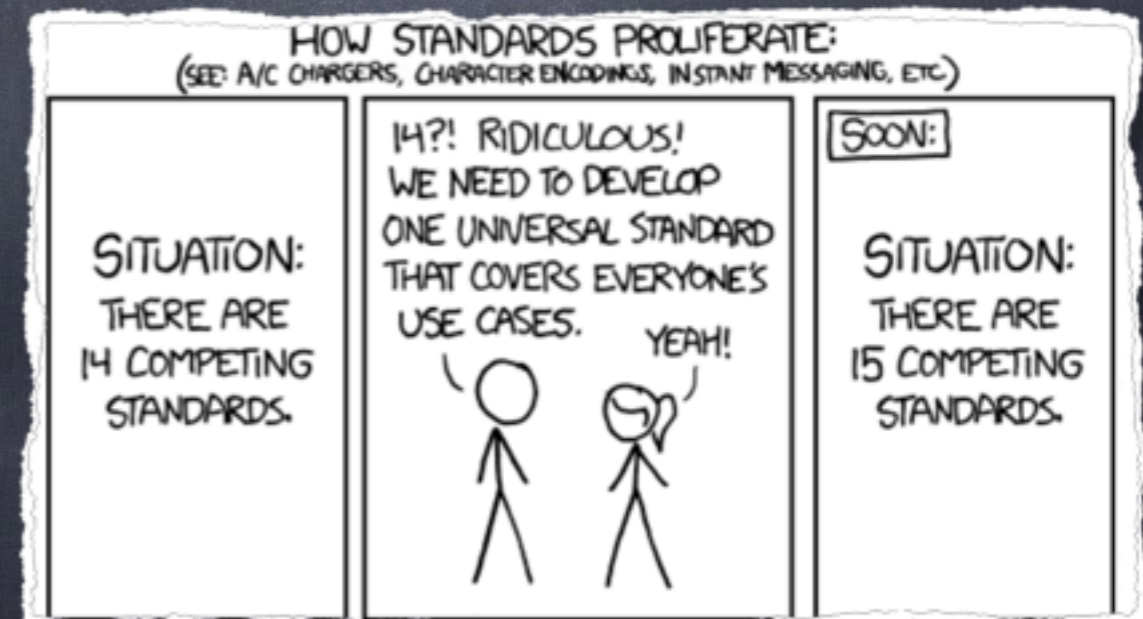
Presented by Giulio Eulisse

- Release process integrated with AliEn
- Maintenance of their own software stack (ROOT/Geant4)
 - ALICE does not foreseen moving to LCG releases
- Build setup
 - Based also in containers as build artefacts NOT as SW distribution mechanism —> CVMFS used (mounted inside the container)
 - Jenkins
 - AliBuild serving the need of the collaboration
 - ALICE will include support to LCG View-like environment

Interaction points with ALICE

- LIM artefacts store
 - Managing logs, tarballs repos
—> common artefact store
- Porting efforts: ARM, Mac
- Common centralise MAC/ARM hardware structure

Giulio Eulisse



THERE IS ANOTHER ISSUE TO RESOLVE: AFS

- Presented by Jan Iven
 - Not a direct issue for SFT/SPI (we are independent)
 - However we need to maintain it for users
 - We agreed on being the connection line with IT bringing this topic to discussion in a regular basis to LIM

AND THEN THE FUTURE: HSF

- Presented by Graeme Stewart

Group's Goals

- Building, packaging and distributing software is a problem faced right across the HEP community (so, not just LHC or even CERN)
 - Every experiment and software group need to put effort into doing this
 - Naively it seems easy, quickly it gets complicated
 - Developers of libraries and toolkits need to care about easy integration into a stack
- So, prima facie, this is an area where we can work together to improve
 - Common build recipes and tools
 - How to take most advantage of technologies like containers
 - Proper support for developers in our collaborations
- Experiment production stacks are vital, but good tools and solutions will be completely portable to other use cases, e.g., lightweight releases for analysis or machine learning

<http://hepsoftwarefoundation.org/activities/packaging.html>

Intersection point: HSF-SFT

- LCGCMAKE is currently providing a cross-community build structure
- source of knowledge and experience

Post-Report

- Checkpoint: Most promising tools seemed to be
 - From our community
 - LCGCMake
 - aliBuild
 - From wider scientific community
 - Spack
- Some prototype work done with Spack to adapt to our use cases
 - Proved Spack community was rather receptive to patches we provided for upstream
- Things went a bit quiet after that
 - Usual case of people being pulled off to other projects...

- Indeed, SPI has made an evaluation effort for Spack thanks to Javier (use in FCC)
- Test drive instructions for the available tools under preparation

Common project with a Summer Student

- Incremental implementation in Spack builds
 - Using LCGCMAKE structure as knowledge source
- 1st supervisor: Javier
- Co-supervisors: Graeme and Patricia (HSF/Spack + SPI/LCGCMAKE)

Some of the actions to take

Internal discussion

- Pre-releases
- Deeper check of the RPMs
- Mattermost setup of LIM
- LCGTests running with some ROOT tests
- Use of the volume in CVMFS
- New build nodes to speed up the releases
- Documentation

Inflexion points topics

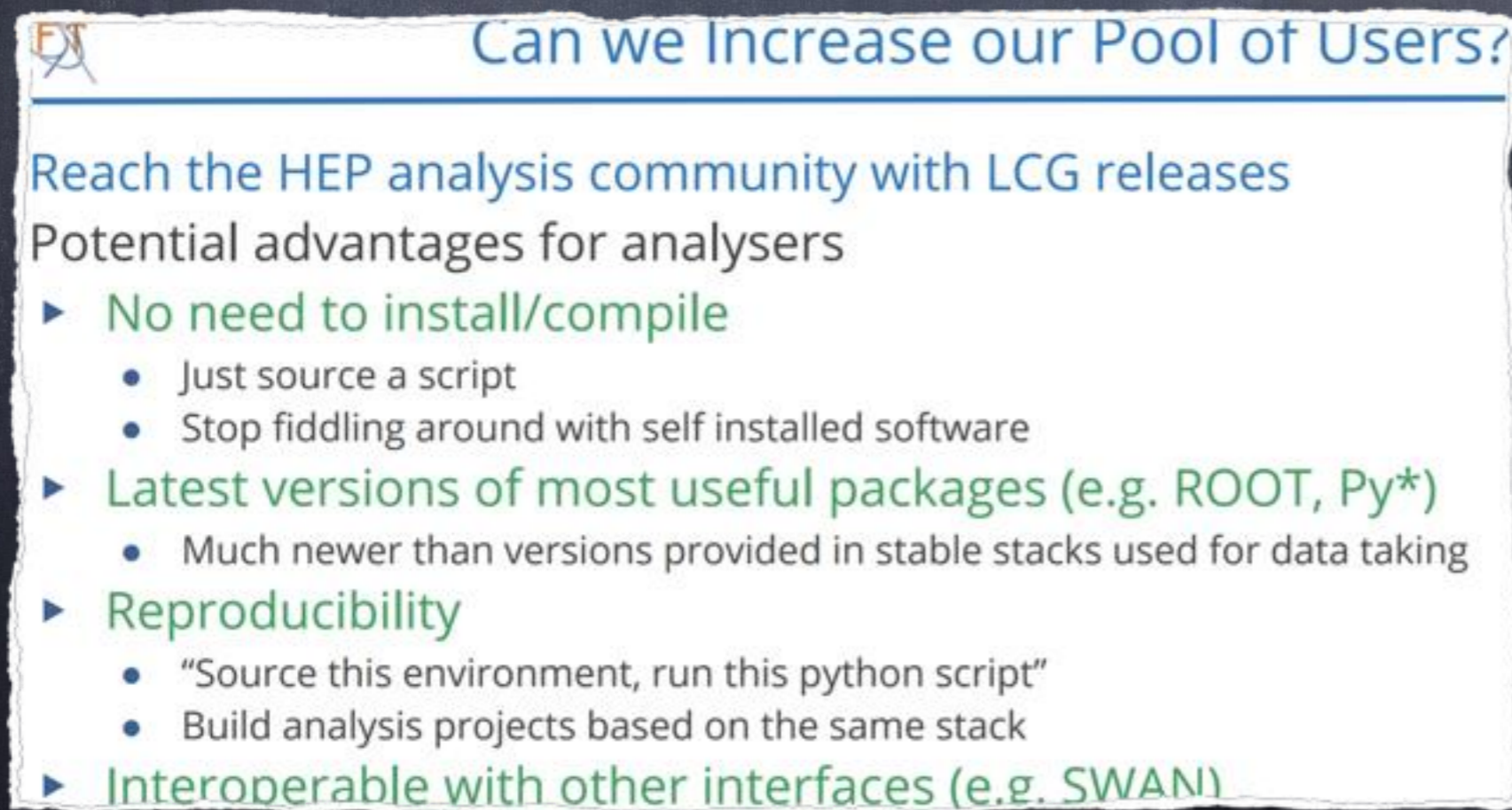
- HEPOS_libs
 - No, we do not want to build their contains even with O(30) extra FTEs, believe me
 - Based on role of SFT as base for externals/builds, can we bring the effort/discussion to this level?
 - HSF is also a good forum of discussion

Common Services

- Proposed by LHCb and ALICE
 - Common Mac, ARM, tarball repos
 - ML ecosystem -> CMS
 - Does it make sense to build it separately?
 - Let me enlarge it: We are maintaining several Jenkins services
 - It is time to come back to IT again?

Danilo' presentation goes in this line

- Enlarge the Pool of users including Analysis presented by **Danilo Piparo**



Can we Increase our Pool of Users?

Reach the HEP analysis community with LCG releases

Potential advantages for analysers

- ▶ No need to install/compile
 - Just source a script
 - Stop fiddling around with self installed software
- ▶ Latest versions of most useful packages (e.g. ROOT, Py*)
 - Much newer than versions provided in stable stacks used for data taking
- ▶ Reproducibility
 - "Source this environment, run this python script"
 - Build analysis projects based on the same stack
- ▶ Interoperable with other interfaces (e.g. SWAN)

Identified set of issues perfectly manageable

MY VERY PERSONAL LAST INPUT

- THIS WAS MY FIRST SLIDE and I am happy to see that for most of the communities we are in the same line



- Unfortunately for just a few we are here

